hepia
Haute école du paysage, d'ingénierie et d'architecture de Genève

Hes·SO // GENÈVE
Haute Ecole Spécialisée de Suisse occidentale

SUPERVISION BASÉE SUR SHINKEN.



THÈSE DE BACHELOR PRÉSENTÉE PAR

Monsieur Huseyin BILGIN
Pour l'obtention du titre Bachelor of Science HES-SO en
Ingénierie des technologies de l'information avec orientation en Logiciel et Systèmes Complexes

Septembre 2015

Professeur HES responsable TB
Gérald Litzistorf

En collaboration avec
Jacques Daudel / Qim info

h e p i a

Haute école du paysage, d'ingénierie
et d'architecture de Genève

Printemps 2015
Session de bachelor

INGÉNIERIE DES TECHNOLOGIES DE L'INFORMATION

ORIENTATION EN LOGICIELS

SUPERVISION BASÉE SUR SHINKEN

Descriptif :

Mise en place d'un outil de surveillance d'un système d'information hétérogène.
Ce système doit permettre de visualiser des alertes ponctuelles et de surveiller les variations de performances des différents systèmes. Il devra surveiller l'ensemble des couches que l'on trouve sur de telles architectures : hardware, systèmes bas niveau, communication/réseau, couches middleware, couches applicatives etc…
Le projet consiste à identifier un sous-ensemble représentatif et de déployer un système qui permet de surveiller ce sous-ensemble.
Le système se limitera à la surveillance et ne devra pas déclencher d'actions correctives.
Il devra être robuste pour assurer la surveillance en toute circonstance et peu intrusif pour ne pas déstabiliser les systèmes de production.

Travail demandé :

Le travail se décompose en plusieurs phases. Il part de l'analyse des besoins à la mise en évidence de la pertinence de la solution proposée.

Nous pouvons distinguer les grandes phases suivantes :

- Inventaire des types d'éléments à surveiller ;
- stratégie du déploiement de la surveillance. Cela peut-être un déploiement horizontal (couche par couche) ou un déploiement vertical (par application métier) ;
- Sélection d'un outil de surveillance avec les arguments prouvant sa puissance, sa robustesse et sa pérennité;
- Identification d'un périmètre pour la réalisation d'un prototype (POC ou Proof Of Concept). Il s'agira certainement de surveiller plusieurs systèmes en environnement UAT (User Acceptance Tests) ;
- Déploiement de la surveillance sur le périmètre sélectionné pour le POC avec un travail particulier sur la représentation des informations ;
- Grandes lignes et estimations pour un déploiement global sur le système d'information.

A la fin du stage nous attendons les 4 livrables suivants, dont un optionnel :

- Un schéma d'architecture ;
- Un POC opérationnel ;
- Un plan de déploiement.
- (option) Des ébauches de procédures de déploiement (ex : comment intégrer la surveillance d'une nouvelle application, d'un nouveau routeur et

Candidat :

M. Bilgin Huseyin

Filière d'études : ITI

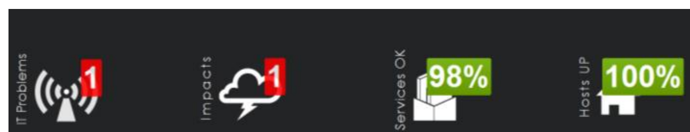
Professeur(s) responsable(s) :
Litzistorf Gérald

En collaboration avec : Qim info

Travail de bachelor soumis à une convention
de stage en entreprise : non
Travail de bachelor soumis à un contrat de
confidentialité : non

Timbre de



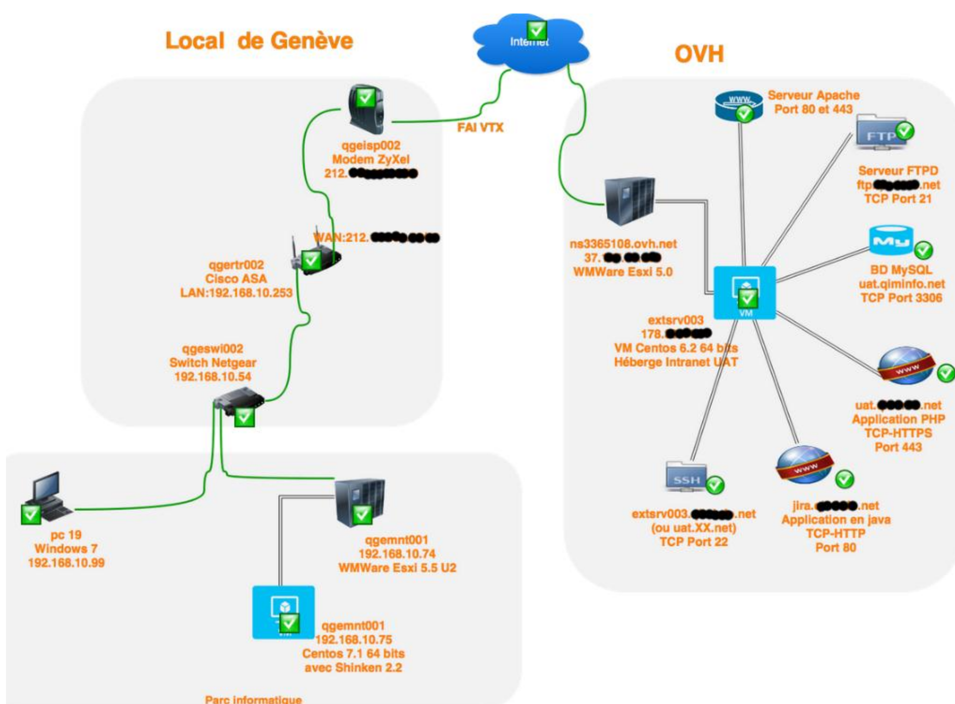


Résumé : Supervision basée sur Shinken

Les infrastructures informatiques deviennent assez complexes et nous n'avons plus de temps pour surveiller manuellement les systèmes. Les administrateurs doivent être aussi proactifs car le core business repose de plus en plus sur le système informatique.

Les solutions de monitoring permettent de remonter des informations techniques et fonctionnelles du système d'information.

Outre cette définition sommaire, ce domaine vaste déborde largement de ce rôle et inclut donc plusieurs activités :



•Surveiller: fonction consistant à connaitre l'état des éléments de la structure informatique.
•Visualiser: présenter sous une forme quelconque l'état global de la structure informatique.
•Analyser: permettre d'établir les rapports voulus sur la santé globale, partielle, particulière.
•Agir: permettre d'être proactif pour éviter les problèmes (Prévenir les défaillances.)
• [Réagir: permettre de corriger automatiquement les problèmes (Relancer des services interrompus)]
•Alerter: permettre d'avertir les personnes concernées qui peuvent ainsi rester passives vis-à-vis de l'outil.

Le but de ce travail est de mettre en place un outil de surveillance d'un système d'information hétérogène. Un scénario de surveillance sera défini pour être inclus dans la supervision. Il faut choisir par rapport aux besoins de l'entreprise parmi plusieurs outils existants.

La supervision définit des métriques pour les éléments supervisés et vérifie plusieurs couches comme matériel ou applicative. Le traitement, grâce aux données récupérées, permet de générer l'historique des pannes et des rapports sur le système surveillé.

Cette mise en œuvre fournit une solution complète qui offre surveillance, métrologie, reporting et cartographie et montre en même temps la faisabilité de la supervision et ses limites.

Candidat :

M. Bilgin Huseyin

Filière d'études : ITI

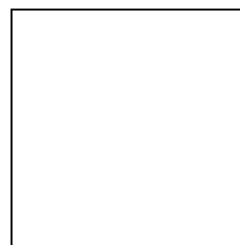
Professeur(s) responsable(s) :
Litzistorf Gérald

En collaboration avec : Qim info

Travail de bachelor soumis à une convention de stage en entreprise : non

Travail de bachelor soumis à un contrat de confidentialité : non

Timbre de



Sommaire

1. Avant-Propos

1.1. REMERCIEMENTS

Je dédie ce travail de mémoire à mon frère Duzgun qui m'a encouragé tout au long de mes études.

Je souhaiterais remercier sincèrement Monsieur LITZISTORF pour ces précieux conseils, qui a toujours suivi mes travaux et m'a orienté dans la bonne direction.

Je remercie beaucoup M Jacques DAUDEL, directeur technique chez Qim info, pour m'avoir donné l'opportunité de réaliser un tel projet dans un cadre professionnel pour se montrer toujours disponible et à l'écoute tout au long de la réalisation de ce travail.

Je remercie également mes collègues de Qim info, surtout M Loïc MALOT chef de projet et M William REY consultant, pour leurs encouragements et pour les merveilleux moments passés ensemble.

Je remercie M Grégoire Hostettler, ingénieur système, pour sa collaboration durant le projet.

Enfin je souhaite remercier spécialement ma chère épouse, Meral, qui m'a aidé de poursuivre mes études jusqu'à aujourd'hui.

Merci à toutes et à tous.

1.2. RÈGLES TYPOGRAPHIQUES

La documentation est faite avec la police Lato Regular 10 pt en suivant les règles typographique des modèles de documents de la Qim Info. Pour certains

- Commande shell : **Lato en gras**
  - **[root@qgemnt001]# /etc/init.d/shinken restart**
  - **[root@qgemnt001]#** signifie utilisateur root
  - **[shinken@shinken]$** signifie utilisateur shinken
  - Certaines commandes sont montrées avec des captures d'écran texte blanc sur fond noir.
  - Si la commande est facultative en Lato normale 9pt

- Code ou paramètres : Sont faits avec des captures texte noir sur fond blanc. Les parties importantes sont surlignées en jaune.
- Chemins de fichiers : *Italique /etc/shinken*
- Les liens de pages web : Lato, bleu et souligné
- Les liens dans le document : Cahier de charges (Lato souligné)
- Les légendes des figures : ** Police de taille 9pt en italique.*
- Mots importants : **En gras.**
- Puce à liste bleu ou verte pour les listes.
- Texte cité depuis une source : *Italique*

1.3. Structure du Rapport

Le rapport est composé de 5 grands chapitres plus les annexes et un avant-propos pour rendre mon travail plus facile à lire et comprendre. Le but principal de la cette structure est de donner la possibilité de mettre en place un superviseur basé sur Shinken tout en apprenant la vision de la supervision.

Pour vous ramener au contexte, je commence avec l'entreprise Qim info et je décris la supervision pour donner un résumé du monde de « Supervision ». Suite à cette introduction je définis le cahier de charges suivant les protocoles utilisés pour la supervision. Les fonctionnalités du système de supervision sont expliquées brièvement.

Le chapitre suivant nous donne l'état de l'art sur les outils de supervision et justifie mon choix qui s'est porté sur Shinken. Ce chapitre aborde également l'architecture et le fonctionnement. J'explique aussi d'autres outils que j'ai couplés avec Shinken.

Le scénario détaille les éléments à superviser, leurs facteurs, les mesures et la stratégie de supervision. Cette partie constitue le cœur du projet en définissant la méthodologie à suivre.
La partie mise en œuvre comprend toute la partie d'installation, et de configuration de Shinken, aussi les outils couplés au système de supervision.

Dans le bilan du projet, j'analyse les résultats obtenus et les tests de validation. Je dénombre les problèmes rencontrés pendant la mise en œuvre et je donne des axes d'améliorations pour que le système devienne plus profitable au sein de l'entreprise.

La partie annexée donne des détails comme les termes techniques, les références ou les procédures techniques. J'explique les étapes de réalisation du projet dans la partie « gestion de projet ».

2. Cahier de Charges

2.1. Contexte



* *Figure 1 Logo de Qim info*

La Qim Info SA a été fondée en 2004 par des personnes issues du domaine du service informatique. Ses locaux sont situés dans la ville de Carouge. Depuis 2010, la société dispose également de locaux dans la ville de Lausanne.

À ce jour, la société compte plus de 120 collaborateurs mais seule une vintaine d'entre eux sont présents dans les locaux. En effet, la plupart des collaborateurs de Qim info sont envoyés pour des missions de plus ou moins longue durée chez le client. Les prestations proposées par la société sont diverses et variées : développement d'applications mobiles, conception de site internet en passant par l'analyse et la maintenance de solutions existantes, etc.

Qim info possède une infrastructure informatique dispersée sur 3 sites : Genève, Lausanne et OVH.

Le local de Genève comprend

- 9 serveurs physiques
- 2 onduleurs
- 1 imprimante réseaux
- 1 parc informatique composé de 19 PC, 3 mac mini, 7 laptops, tablettes ,1 tv Samsung, 1imprimante
- 3 switch ,2 routeurs, 2 modems
- 2 points de sortie vers le réseau internet avec 2 différents fournisseurs d'accès (Swisscom et VTX)
- 3 machines dédiées pour les unités de stockage
- borne wifi
- une centrale téléphonique IP PABX

Le bureau de Lausanne comprend :

- 1 routeur
- 1 imprimante
- 2 PCs

Le site distant chez OVH :

- 1 serveur physique avec Hyperviseur Esxi qui héberge 3 VM (Machine Virtuelle) avec CentOS
- 1 Serveur physique de backup FTP
- 1 VM avec hébergement Windows pour une application web
- 1 VM avec centos comme serveur APNS

Tous les employés utilisent les ressources et les services informatiques constamment. Les produits fabriqués (logiciels, applications) sont hébergés sur les serveurs. Ces serveurs hébergent aussi les applications et les services nécessaires pour faire tourner les activités. A l'intérieur des locaux, le réseau LAN regroupe l'infrastructure et 2 points de sorties garantissent l'accès au réseau internet (providers Swisscom et VTX). Comme une importante infrastructure est hébergée chez OVH, l'accès à celui-ci doit toujours être opérationnel.

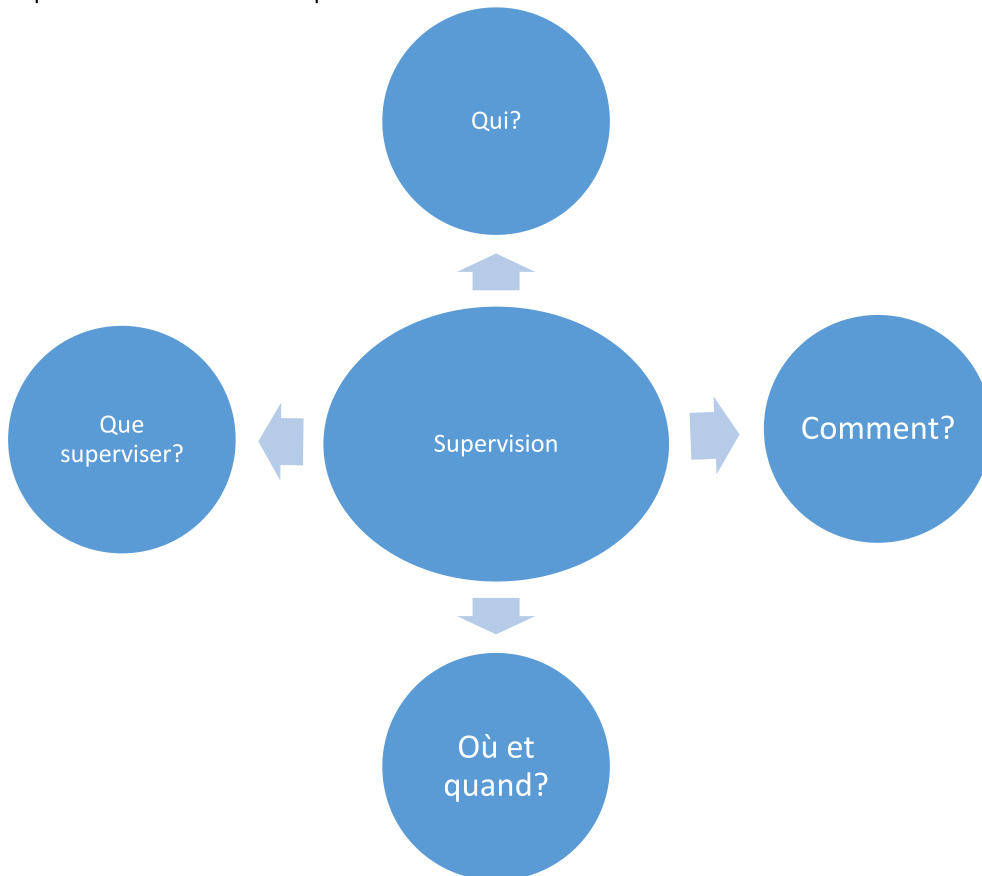
Le business dépend fortement de la fiabilité du système informatique. Nous devons être en mesure de détecter toutes les pannes survenues sur ce système, surtout celles qui peuvent empêcher les activités normales (Business as Usual). Ces pannes peuvent avoir un coût important s'il n'y pas d'intervention. Mais la détection d'une panne lors de son apparition ne suffit pas car seul un système prédictif peut empêcher une panne.

2.1.1. ***Problématique***

Actuellement la surveillance est assurée par l'intervention de M Hostettler (Ingénieur système) qui est contrôlé par M Daudel (Directeur Technique) et M Malot (Chef de projet) d'une façon non automatisée. Cette manière ne suffit plus aux besoins de l'entreprise.
La nécessité d'un système de surveillance automatisé est vitale. Ce système doit détecter certaines anomalies en avance et ne doit pas nuire aux performances du système. Mais il doit être aussi adaptable pour une infrastructure distribuée et hétérogène.

2.2. SUPERVISION

La figure ci-dessous m'a permis de réfléchir sur la méthodologie et la technique à utiliser pour la supervision. Elle résume les parties à étudier :



** Figure 2 Mind map pour la supervision*

2.3. QU'EST-CE QUE LA SUPERVISION ?

Depuis quelques années la surveillance des infrastructures informatiques est devenue incontournable. Cette manière de surveiller apporte la possibilité d'avoir une vue globale sur l'état de l'infrastructure mais aussi sur les possibilités d'auditer les systèmes, surveiller la disponibilité et les performances et alerter les responsables.

Une supervision consiste à mesurer divers métriques permettant de renseigner les responsables sur la qualité d'un service. Un système de supervision est composé de fonctions qui consistent à mettre en place des indicateurs d'état sur une infrastructure informatique.

La supervision peut permettre à une entreprise de :

- Vérifier l'état d'un périphérique ou d'un service
- Remonter des alertes
- Détecter des anomalies et pannes



** Figure 3 Chaîne de la supervision*

2.4. QUE SUPERVISER ?

La pondération des éléments surveillés dépend de l'importance d'un élément pour le business. Cela veut dire que certains machines, serveur ou d'autre élément du système informatique peuvent avoir un effet plus importance que les autres concernant la productivité de l'entreprise. Exemple : si un routeur tombe en utilisant le 2ème routeur on peut facilement continuer à travailler, par contre si c'est le serveur d'intranet ou un serveur de production sur lequel on héberge les applications importantes qui tombe, cela peut perturber le travail des développeurs ou administration.



Supervision peut être décomposé en 4 grandes parties ;

- La supervision technique (système) :

Elle va consister à surveiller le réseau, l'infrastructure et les machines du système d'information

- La supervision applicative :

Cette partie consiste à surveiller les applications.

- La supervision métier :

Consiste à surveiller les processus métiers.

- La supervision de la sécurité :

Surveillances des attaques contre le système informatique.

** Figure 4 – Couches des éléments pour la supervision*

2.5. POURQUOI SUPERVISER, LES ENJEUX?

L'infrastructure informatique comprend de nombreux systèmes qui peuvent mal fonctionner. Si un problème n'est pas détecté au bon moment, il peut engendrer des sérieux dégâts à l'entreprise vu que le business dépend fortement du système informatique. Avec un superviseur nous avons la possibilité d'anticiper une panne et la résoudre plus rapidement. Cela permet aussi de surveiller la santé du système.

Prévenir en cas de problème et réduire les délais et coût des interventions sont les enjeux principaux. Une bonne supervision aide aux responsables informatiques pour rendre des comptes, être au courant et corriger QoS (Quality of Service).

2.6. COMMENT SUPERVISER ?

Le principe de supervision implique l'utilisation de beaucoup de services et cela est complexe : le superviseur teste les services et les équipements avec des outils d'acquisition et il va alerter le responsable en cas de panne.

Il fournit également une interface pour surveiller le système en temps réel. Ces tests se font via l'exécution de certains binaires ou scripts qui se trouvent sur le superviseur.

Ces tests qu'on appelle sonde (check) se font distinguer par deux approches : avec ou sans un agent. La différence de ces deux méthodes repose sur l'installation nécessaire sur l'élément à superviser. Les sondes sans agent comme des sondes tcp, http, ssh se font souvent sans rien installer sur l'élément. Mais dans le cas d'une sonde avec ssh, si l'élément n'a pas de serveur ssh, il faut l'installer. Alors que dans le cas d'une sonde via SNMP ou encore nrpe il faut faire des installations et des configurations. Dans mon cas, vu le nombre de machines surveillées, je n'ai pas beaucoup d'installation à faire mais pour un système plus volumineux ces installations peuvent créer des soucis. Les problèmes de sécurité s'ajoutent car chaque protocole oblige l'ouverture de certains ports sur les firewalls.

Pour une sonde en SSH la clé publique de shinken doit se trouver aussi dans les clés autorisées de cette machine. L'avantage de cette méthode est aussi la facilité d'automatiser le processus de déploiement de la clé SSH avec un script ou Snake oïl.

2.6.1. Comparaison de deux sondes (HTTP et SNMP)

La manière de récupérer les données pour les sondes (check) varie beaucoup selon les méthodes utilisées. Par exemple dans le cas d'une sonde en http, l'outil fait tout simplement une requête http vers une url et analyse la réponse reçue pour déterminer l'état de service.

Dans le cas d'une sonde snmp, on fait une requête snmp (encapsulée dans UDP) vers une machine surveillée. Cette machine doit avoir le service snmp activé pour écouter les requêtes snmp depuis l'extérieur. Dès la réception d'une requête la base de données des objets (MIB) est consultée. Le service peut envoyer une information demandée avec son identifiant (OID) ou toutes les informations qu'il possède.

a) Exemple sondes http en perl :

```
my $url = $protocol . "://" . $o_host . ":" . $o_port . $o_url ;
my $request = HTTP::Request->new('GET', $url);
my $response = ..->request($request);
# Get the HTTP Responde Code (200, 404, 500 ...)
my $code = $response->code();
my $content = $response->content();
if($code ....)
```

b) Exemple sondes SNMP en perl :

```
#Connexion avec la machine
foreach $oid (@oids)
($session,$error) = Net::SNMP->session(Hostname => $host,
                        Community => $community, Port => $port);
$get = $session->get_request("$oid");
$session->close;
$value = $get->{$oid};
#recuperation des données
push @{ $results{$label} },($value,$warn,$crit,$v,$an);
```

Le monde de la supervision possède des standards et la supervision est faite via des protocoles. Les principaux protocoles sont :

2.6.2. SNMP – Simple Network Management Protocol

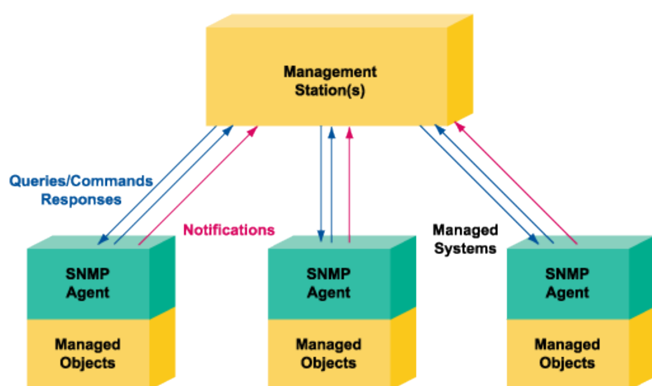
Ce protocole est plus utilisé dans le monde de supervision, la première version (SNMP v1) date de 1987, mais la tendance est d'abandonner le SNMP et d'aller vers d'autres outils qui conviennent mieux aux besoins du monde informatique comme la supervision applicative. Ce protocole est encapsulé dans UDP et la communication se fait via les ports 161/162.

Snmp est un protocole de communication pour superviser et diagnostiquer les équipements de réseau. Il permet principalement de :

- Fournir des informations détaillées concernant les équipements et le réseau
- Paramétrer les équipements du réseau
- Alerter les responsables

Beaucoup d'outils de supervision utilisent SNMP pour avoir l'état des équipements. (Centrerons, Net Crunch 5, MRTG, Cati, Schinkel, Nagios, Zarbi).

Un agent installé sur l'équipement géré transmet les données au format SNMP. Un nom de communauté est défini pour la communication. Une restriction est possible au niveau des adresses IP pour communiquer avec l'extérieur. Les responsables peuvent gérer et visualiser les équipements depuis une centrale. SNMP obtient les informations depuis une base de données depuis MIB (Management Information Base) mais malgré que MIB contienne beaucoup de données, one ne peut pas obtenir les informations de haut niveau comme état d'un service http ou autre.



* *Figure 5 Schéma du fonctionnement SNMP*

2.6.3. IPMI – Intelligent Platform Management Interface

L'Interface de gestion intelligente de matériel, est un ensemble de spécifications d'interfaces communes avec du matériel informatique (principalement des serveurs) permettant de surveiller certains composants (ventilateur, sonde de température, ...), mais également de contrôler l'ordinateur à distance, reboot, interrupteur, console à distance.

2.6.4. CIM- Common Interface Model

Le standard ouvert pour définir comment représenter les éléments administrés sous forme d'un ensemble unifié et cohérent.

2.6.5. JMX – Java Management Interface

API pour la gestion de fonctionnement d'un logiciel java.

2.6.1. WBEM – Web Based Enterprise Management

L'ensemble de standards pour unifier la gestion des infrastructures.

2.6.2. SBLIM- Standard Based Linux Instrumentation for Manageability

Permet d'utiliser les technologies de WBEM depuis des machines Linux.

2.6.3. WS-MANAGEMENT - Web Services for Management

Utilise services web basés sur SOAP pour la gestion.

2.6.4. WMI - Windows Maganement Instrumentation

C'est une implémentation du Web-Based Entreprise Management pour les plates-formes Windows qui étend le modèle CIM (Common Information Model). Elle est puissante au niveau de ses fonctionnalités.

2.7. FONCTIONNALITÉ

2.7.1. Objectifs

Mise en place d'un outil de surveillance d'un système d'information hétérogène.
Ce système doit permettre de visualiser des alertes ponctuelles et de surveiller les différents systèmes.
Il devra surveiller l'ensemble des couches que l'on trouve sur de telles architectures : hardware, système bas niveau, communication réseau, couches middleware, couches applicatives (sur tous les trois sites existants).
Les paramètres des éléments de l'infrastructure à surveiller :
Charge CPU
Charge RAM
Quota de disque
Bande passante
Etat de l'hôte (Réponse au Ping etc.)
Etat des services (web, SQL etc.)
Le système se limitera à la surveillance et ne devra pas déclencher d'actions correctives.
Il devra être robuste pour assurer la surveillance en toute circonstance et peu intrusif pour ne pas déstabiliser les systèmes de production.
Un affichage en temps réel aidera les administrateurs à résoudre le problème après avoir reçu l'alerte en aide au diagnostic.

2.7.2. Fonctions de surveillance

- Serveurs
  - Charge (CPU, RAM, disque, réseau), à surveiller avec un seuil
  - État des services
- Applications, services hébergées par les serveurs
  - Etat de services
- VMware avec Hyperviseur ESXi
  - État de l'hôte : Charge (CPU, RAM, Disque, Réseaux)
- Machine virtuelle
  - Charge (CPU, RAM, Disque, Réseaux), à surveiller avec un seuil
- Postes de travail : Pc, Mac, laptops
  - Allumé ou pas, pour certains
  - Charge (CPU, RAM), à surveiller avec un seuil
  - Quota disque, à surveiller avec un seuil

- Switch
    - Disponibilité
- Routeurs
    - Disponibilité
    - Accès vers l'extérieur
- Modem
    - Disponibilité
    - Accès vers l'extérieur
- Imprimante
    - Disponibilité
- Borne d'accès Wifi
    - Disponibilité

2.7.3. ***Fonctions de communication***

Le système envoie un mail à une liste de distribution it@qiminfo.ch avec un message clair contenant la source de panne, le degré et le plus de détails possibles.
Le système doit envoyer des notifications aux responsables. (D'autres méthodes de communication peuvent être implémentées dans le futur, choisir le responsable selon la panne et l'horaire peut être une possibilité aussi)
Niveau des alertes
Le système de surveillance remonte une alerte mais des fois cela n'empêche pas le blocage du service informatique. Le système peut être prédictif pour certains cas et lancer des alertes avant la production de l'erreur. Exemple : lancer une alerte quand le quota de disque atteint à 90% au lieu de lancer l'alerte à 100%. Ce type d'avertissement peut avoir un niveau moins important mais facilite le travail de l'administrateur.
Il faudra aussi définir les seuils d'utilisation acceptables pour déclencher des alertes quand on s'approche de la limite (ex : CPU utilisée pendant plus de 5 minutes à plus de 80%).

2.7.4. ***Fonction d'affichage***

Le système de supervision affiche en temps réel l'état des systèmes avec plusieurs vues. On peut afficher une vue globale et des vues détaillées.
Les administrateurs veulent voir l'état général du système et ils ont la possibilité de choisir les éléments à visualiser. Nous pouvons par exemple vouloir afficher l'état de système par couche. Ex: afficher juste les serveurs web ou les quotas de disques.

Interface web personnalisé pour surveiller l'infrastructure en temps réel. Créer plusieurs vue éventuellement une première vue générale qui résume l'état général du système.

Proposer des vues détaillées pour les éléments à surveiller.

3. Etude et Choix de l’Outil

3.1. Critères d’études

Les critères de choix se basent sur le cahier de charge définie dans la section2.7. N’ayant pas d’assez de temps pour installer et essayer ces outils, je me suis basé sur les études et les comparaisons existantes avec les spécifications fournies.

	Comptabilité avec VMware	Paramétrage, Modularité, ajout des plugins	Déploiement sur système distribué et hétérogène	Réactivité de la communauté et équipe de développement	Fonctionnalité de supervision (quels équipements et quels facteurs sur ces équipements)	Performance	Interface web, vue custom	Platform (Win/Lin)
Nagios	+	-+++	++	+-	+++	++ (rapide car écrit en C)	++ (interface de base)	+ (Linux)
Shinken	+	-+++	+++ (idéal pour distribuer)	++	+++	+++ (efficace car optimisé)	++ (plus moderne)	++
EON Eyes Of Network (nagios+cacti+..)	+	+-++	++	-+	+++	+ +	++	- (os dédié centos)
Centreon (Nagios)	+	-++	++	++	+++	+ +	++	
ZABBIX	+	-+-	++	++	+++	++	++	++
openNMS	+	-+-	++	-+	++	+	+-	++
VIGILO (Nagios)	+	-+-	++	--	+++	+	+ +	+ (Linux)

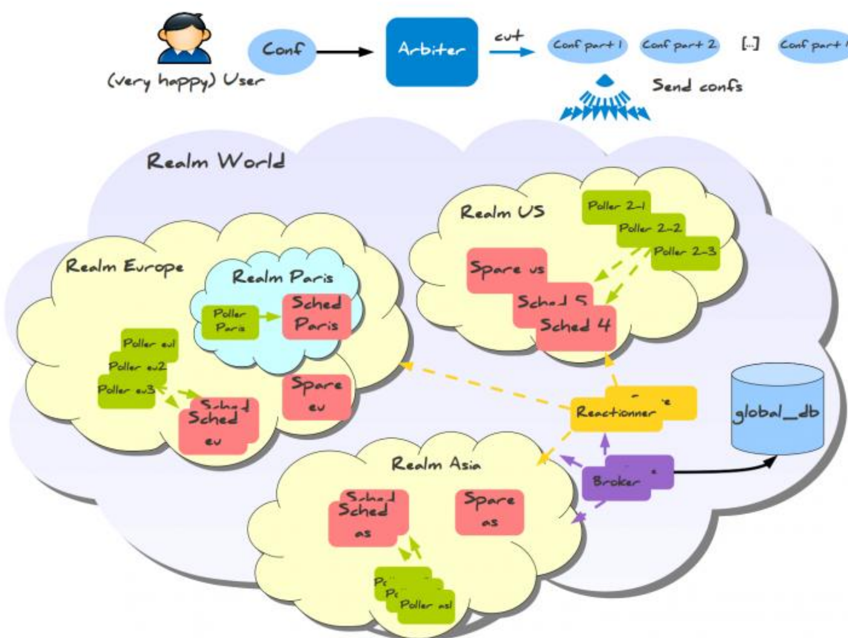
** Tableau 1 Tableau de comparaison des outils*

3.2. CONCLUSION DE L'ÉTUDE

Parmi les outils étudiés Nagios et Shinken sont les plus adaptés au projet.
Interface : Shinken possède une interface plus moderne et l'interface de l'utilisateur marche sans un serveur apache ou autre.
Performance : Malgré que Nagios soit écrit en C, Shinken fournit des performances améliorées par rapport à Nagios.
Le nombre max de sondes en 5minutes.

- Nagios 3 (C) <=> icinga 1.6 (C) = 30K
- Centreon-engine (C) = 25K
- Shinken = 120K

Source : Jean Gabès http://fr.slideshare.net/JeanGabs/confrence-shinken-sophiaconf2012-jean-gabs
Modularité : Shinken est totalement modulaire avec système de multiples démons.
Système distribué : Shinken est développé surtout pour permettre de superviser un système distribué. Sa configuration est plus simple que Nagios. Il donne aussi la possibilité du degré de distribution. Nous pouvons configurer tous les sites facilement avec une seule configuration et choisir les parties à déployer dans les sites distants.
Business rule(Règles métier) : Shinken introduit la règles métier dans ses implémentations ce qui est bénéfique pour les sociétés services informatiques.
Plateforme : Shinken peut également tourner sous Windows alors que Nagios ne peut pas.
Mais Nagios comporte quand même des avantages comme des modules binaires déjà compilés et une interface détaillée.
Shinken se différencie avec sa modularité et adaptation pour un système distribué. Il est aussi optimisé pour une meilleure performance. Son amélioration au niveau des performances est nettement considérable auprès Jean Gabès.
Malgré que Shinken soit nouveau par rapport à Nagios, il a une communauté très réactive. Shinken possède au moins un millier de contributeurs et cela ne cesse d'augmenter.
Finalement mon choix se porte sur Shinken après cette comparaison de critères. La figure 6 ci-dessous illustre bien l'architecture de Shinken sur un système distribué qui convient aussi à l'entreprise Qim info.



La présentation de M Sébastien Pasche, à la date du 11 mai 2015, qui est ingénieur de système chez le shop nous a bien confirmé tout ceci. Contributeur de Fedora, Shinken et plusieurs projets d'open source M Pasche pense aussi que Shinken évolue très rapidement et Nagios subit beaucoup et il ne change plus depuis plusieurs années.

* *Figure 6 Exemple d'architecture Shinken pour un système distribué*

Source : http://shinken.readthedocs.org/en/latest/07_advanced/distributed-with-realm.html

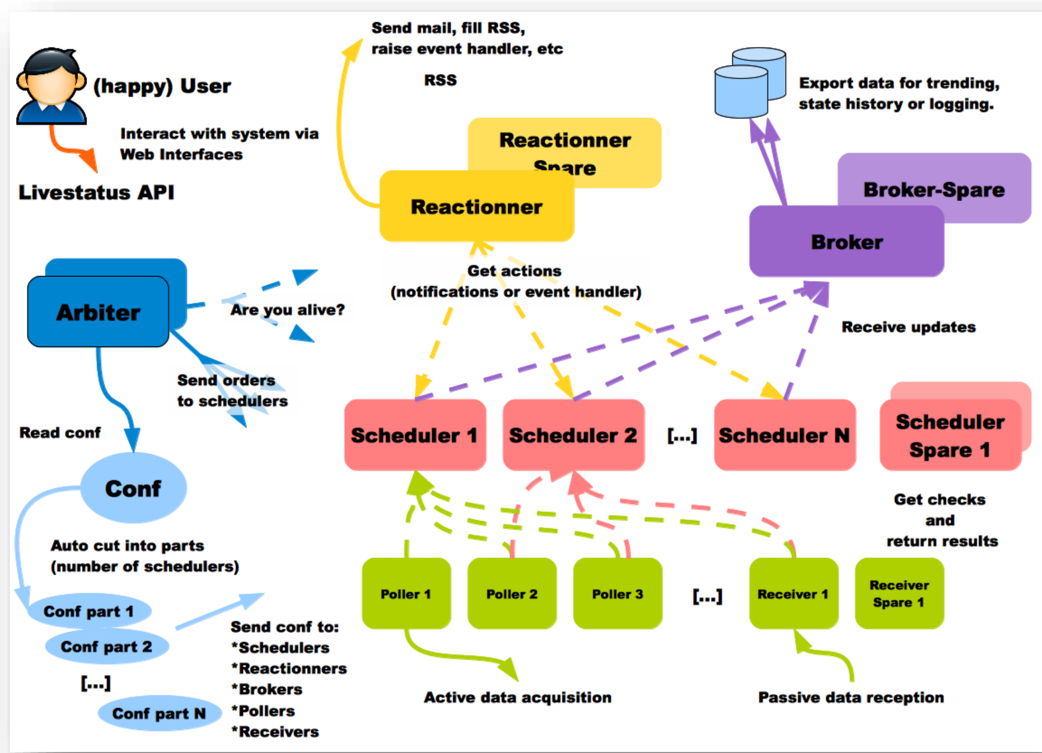
3.3. Shinken



* Figure 7 Logo Shinken

« Shinken est un outil de supervision, une implémentation de Nagios en Python 'from scratch' qui a pour objectifs principaux de simplifier la vie des administrateurs de grands parcs et de coller au mieux à l'évolution de l'IT des dix dernières années, et si possible préparer les dix prochaines... »
Jean Gabès, le créateur de Shinken lors d'une interview en 2011.

Le projet Shinken a commencé avec un fork de Nagios par Jean Gabès pour corriger certains bugs qu'il avait découverts. Il réécrit en Python en obtenant des résultats excellents surtout au niveau des performances et avertit l'équipe de Nagios mais l'équipe de Nagios n'acceptait pas les changements.
Alors le Shinken est né comme un petit frère de Nagios en 2009 pourtant il a très vitre évolué grâce aux nombreux contributeurs. L'idée principale était de découper le cœur monologique de Nagios en plusieurs parties par rapport aux différents rôles.
Shinken révolutionne l'architecture de Nagios mais également apporte des solutions pour la scalabilité, la supervision des impacts de business. Son découpage fonctionnel nous permet d'adapter l'outil sur une infrastructure distribué. Il offre une haute disponibilité et la répartition de charges.
Cet outil facilite aussi la détection de la source des incidents et éviter des fausses alertes.
Il a été développé pour devenir une solution complète de supervision mais à l'heure actuelle il est considéré comme un framework de supervision. Et l'équipe préfère l'intégration des outils comme les outils de métrologie à la place de tout développer dans le projet Shinken.
L'utilisation de Shinken est sous licence GNU (AGPL- Licence publique générale Affero).



* Figure 8 Architecture Shinken

Source : http://blog.nicolargo.com/2012/11/installation-pas-a-pas-dun-serveur-de-supervision-shinken.html/shinken-architecture

3.3.1. Architecture

Shinken a mise beaucoup sur son architecture qui le différencie des outils de supervision. Il est composé de six démons (daemon -service) et sa modularité repose sur ce principe. Chaque démon a son rôle. À part l'unique démon « arbiter », nous pouvons multiplier et déployer les démons à l'endroit voulu. Cet endroit peut être un site distant un le DMZ(Demilitarized Zone) qui convient mieux par rapport à l'architecture de l'infrastructure. Cette modularité supporte des grandes charges et fournit la haute disponibilité.

a) Arbiter

Ce démon unique a comme rôle de gérer tous les autres démons. Il lit la configuration, la découpe, et l'envoie aux éléments de l'architecture. Il gère aussi la haute disponibilité : si un élément meurt, il envoie la configuration à un autre démon disponible.

b) Scheduler

Il ordonnance les tests et gère des actions en cas de soucis lors des sondes. Son rôle n'est pas lancer les sondes mais de planifier l'heure des sondes pour le Poller et récupérer les résultats de ces tests pour envoyer au Reactionner et Broker. On peut en avoir autant qu'on veut, l'Arbiter découpe la configuration par rapport au nombre de schedulers définis comme dans la figure 8 ci-dessus.

c) Poller

Le rôle d'un Poller est lancer une sonde en exécutant un binaire ou un script mais aussi retourner les résultats au Scheduler. On peut en avoir autant qu'on en a besoin. Pour des systèmes distribués, volumineux ou complexe les Pollers peuvent être positionnés dans différents endroits pour contourner les problèmes de performances et sécurité.

d) Reactionner

Le Reactionner gère l'envoie des notifications selon les directions de Scheduler. Souvent il envoie des mails mais aussi il peut envoyer des sms via des modules.

e) Broker

Il est chargé de récupérer les données depuis le Scheduler et de les mettre dans d'autres outils via des modules. Par exemple il peut exporter des données de métrologie vers Graphite ou PNP4Nagios via le module Livestatus. L'interface web de Shinken récupère les informations depuis le broker via le module Webui.

f) Receiver

Le seul démon facultatif, utilisé pour recevoir les données de sonde passif qui est lancé par un élément surveillé contrairement à une sonde actif lancé par un Poller.

3.3.2. Plugins

Un plugin est un exécutable, peu importe le langage de codage. Cela peut être un binaire compilé ou un script. Tant que ce plugin envoie le bon code de retour, il est compatible avec Shinken. Le plugin peut envoyer des données de type « performance data » pour être traités avec des outils externes. Si le plugin envoie de données, le code de retour doit être séparé avec « | ». Le plugin peut envoyer plusieurs lignes de données. C'est finalement plugin qui indique au Shinken l'état de l'élément surveillé avec quatre types de code de retours basé sur POSIX. Le retour de l'un de ces est obligatoire.

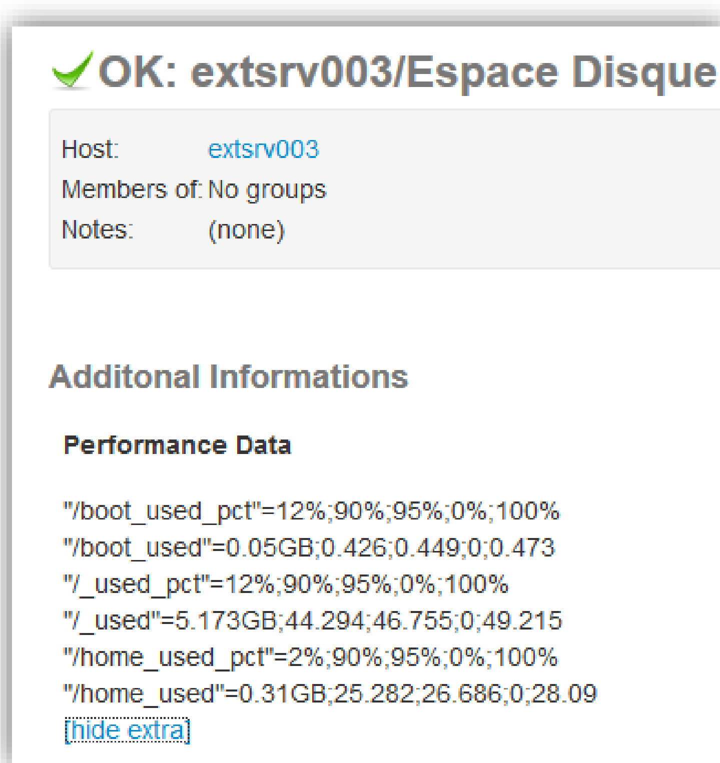
- 0 - OK : Tout va bien.

- 1 – Warning : Risque potentiel problème (Les métriques testés ont atteint le niveau warning).
- 2 – Critical : Problème survenu.
- 3 – Unknown : Le check n'a eu lieu.

La plupart de ces plugins sont écrit en Perl et Python.

```
[shinken@qgemnt001 shinken_backup]$ /var/lib/shinken/libexec/check_ntp_time -H 192.168.10.74
NTP CRITICAL: No response from NTP server
[shinken@qgemnt001 shinken_backup]$ /var/lib/shinken/libexec/check_disk -w 15% -c 5% -p /dev/sda2
DISK OK - free space: /dev 1886 MB (100% inode=99%);| /dev=0MB;1603;1791;0;1886
[shinken@qgemnt001 shinken_backup]$ /var/lib/shinken/libexec/check_ssh_connexion.py -H 178.[redacted]7
OK: connexion is good
[shinken@qgemnt001 shinken_backup]$ /var/lib/shinken/libexec/check_dns -H uat.qiminfo.net -a 178.[redacted]7
DNS OK: 0.070 seconds response time. uat.qiminfo.net returns 178.[redacted]7|time=0.069768s;;;0.000000
```

* *Figure 9 Exemple du résultat des plugins dans Shell*



* *Figure 10Performance data retourné par le check disk via SSH sur interface Web*

Les données retournées sont gardées dans trois macros (variables).

$SERVICEOUTPUT$	DISK OK - free space: / 3326 MB (56%);
$SERVICEPERFDATA$	/=2643MB;5948;5958;0;5968"/boot=68MB;88;93;0;98"/home=69357MB;253404;253409;0;253414"/var/log=818MB;970;975;0;980
$LONGSERVICEOUTPUT$	/ 15272 MB (77%);n/boot 68 MB (69%);n/var/log 819 MB (84%);

Shinken peut accepter au maximum 8KB de données par défaut mais on peut le modifier via la variable max_pluginsnet_output_length.
Ces plugins sont trouvables sur le site monitoring plugins mais aussi on peut utiliser les plugins de Nagios qui sont disponibles sur le portail de Nagios.

3.3.3. ***Modules***

Les modules sont ajoutés pour étendre la fonctionnalité de supervision. Ils sont intégrés dans les démons tels que Broker ou Arbiter. Chaque module doit être définie dans Shinken et lié au démon concerné.
Certains modules comme Livestatus, webui ou encore les modules de log sont des indispensables pour Shinken. Le module webui permet d'avoir l'interface web de shinken native. Un module de log est nécessaire pour le système log.

```
## Module:      webui
## Loaded by:   Broker
# The Shinken web interface and integrated web server.
define module {
    module_name          webui
    module_type          webui
    host                 0.0.0.0     ; All interfaces = 0.0.0.0
    port                 7767
    auth_secret          [illegible]*     ; CHANGE THIS or someone could forge cookies
    allow_html_output    1       ; Allow or not HTML chars in plugins output.
                                 ; WARNING: Allowing can be a security issue.
    max_output_length    1024     ; Maximum output length for plugin output in webui
    manage_acl           1       ; Use contacts ACL. 0 allow actions for all.
    play_sound           0       ; Play sound on new non-acknowledged problems.
    login_text           Welcome on Shinken WebUI    ; Text in the login form.

    ## Modules for WebUI
    # - auth-htpasswd         = Use an htpasswd file for auth backend.
    # - auth-active-directory = Use AD for auth backend (and retrieve photos).
    # - auth-cfg-password     = Use the password setted in Shinken contact for auth.
    # - ui-pnp                = Use PNP graphs in the UI.
    # - ui-graphite           = Use graphs from Graphite time series database.
    # - mongodb               = Save user preferences to a Mongodb database
    # - SQLitedb              = Save user preferences to a SQLite database

    #pour web ui
    modules auth-cfg-password,auth-active-directory,SQLitedb,Graphite-UI,livestatus

    ## Advanced Options
    # Don't play with them on your production server ;)
    #http_backend            auto     ; Choice is: auto, wsgiref or cherrypy if available
    #remote_user_enable      1        ; If WebUI is behind a web server which
                                      ; has already authentified user, enable.
    #remote_user_enable      2        ; Look for remote user in the WSGI environment
                                      ; instead of the HTTP header. This allows
                                      ; for fastcgi (flup) and scgi (flupscgi)
                                      ; integration, eg. with the apache modules.
    #remote_user_variable    X_Remote_User  ; Set to the HTTP header containing
                                      ; the authentificated user s name, which
                                      ; must be a Shinken contact.
    # If you got external plugins (pages) to load on webui
    #additional_plugins_dir

}
```

** Figure 11 déclaration d'un module dan /etc/shinken/modules*

Suite à la déclaration du module, il faut aussi ajouter dans le démon concerné. En l'occurrence pour l'interface web, c'est le démon broker qui s'en occupe pour fournir les données.

```
#activations des modules,interface web
 modules webui,livestatus,Graphite-Perfdata,Canopsis
```

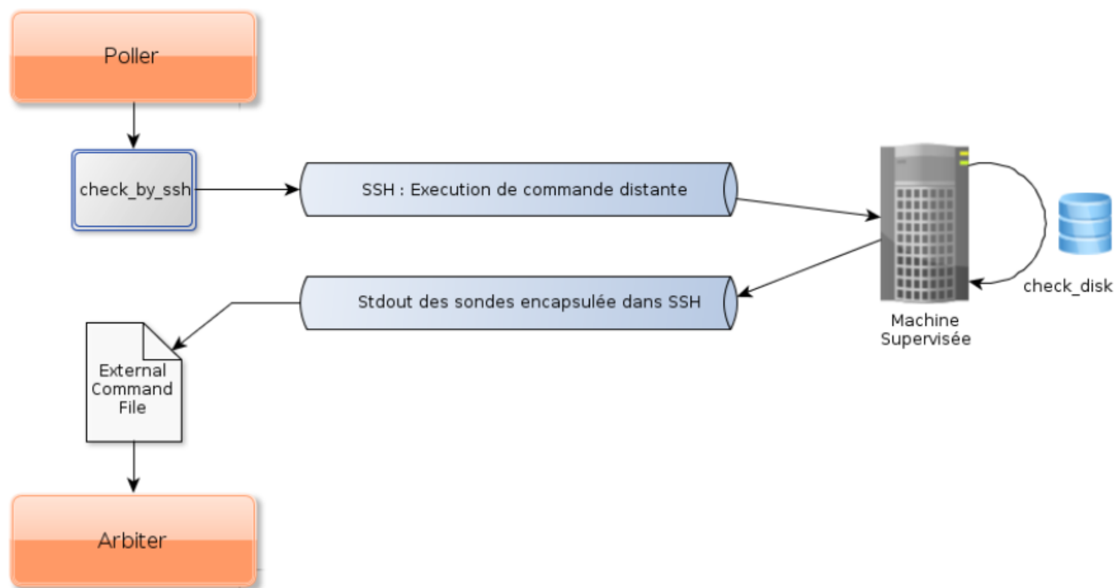
** Figure 12 Ajout du module webui dans le démon broker*

3.3.4. ***Check (sonde)***

Un Check est la base d'un système de supervision car ce sont les sondes qui font le travail. Une sonde est une vérification d'un élément à surveiller. Cela peut être un Ping pour tester un host ou une requête http afin de vérifier la disponibilité d'une URL. Une sonde est utilisé dans la configuration d'un host ou service vie une commande. La commande détermine utilisation d'un plugin avec les paramètres à passer à l'exécutable.

Shinken, comme Nagios, définit 4 niveaux de message de retour pour les sondes. Pour une vérification (check) il faut définir des niveaux pour warning et critical. Ces niveaux peuvent être ajustés au fur et à mesure. Dans la possibilité je privilégie des check par SSH. Une sonde en SSH est sécurisée et faite sans installer un agent sur l'élément à surveiller. Mais il faut envoyer la clé SSH publique du serveur de monitoring vers l'hôte surveillé. Pour chaque type d'éléments vous avez les facteurs à surveiller, les méthodes et les niveaux. En cas d'impossibilité de sonde SSH, deuxième méthode sera des sondes par SNMP. Concernant les machines avec Windows des sondes via WMI sont prévus car cette méthode se fait sans installation spécifique sur la machine monitoré. Cela consiste à récupérer les informations depuis la gestion interne Windows avec un utilisateur existant. Donc il faut préalablement avoir un compte pour le Shinken dans l'AD. Une troisième catégorie de sondes est http qui est très utiles pour la supervision applicative. Par rapport à l'application supervisée il y a des check basé en tcp ou udp pour tester les services comme MySQL, FTP ou DNS.

Dans notre cas les sondes sont en mode actif, cela veut dire que c'est le serveur de monitoring qui interroge les éléments à surveiller alors qu'en mode passif c'est contraire.



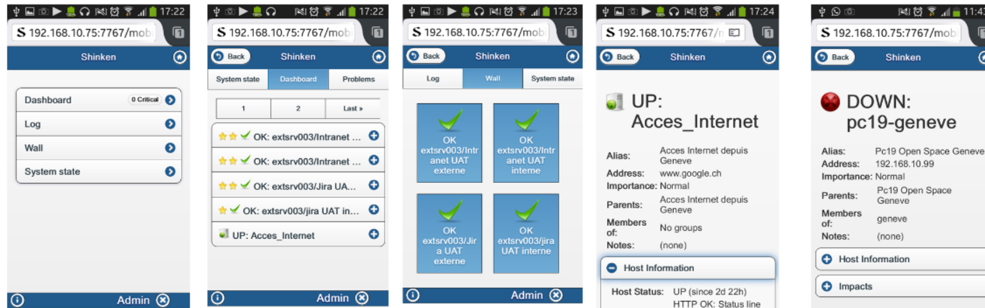
** Figure 13 Fonctionnement d'un check en SSH*

3.3.5. ***Agent***

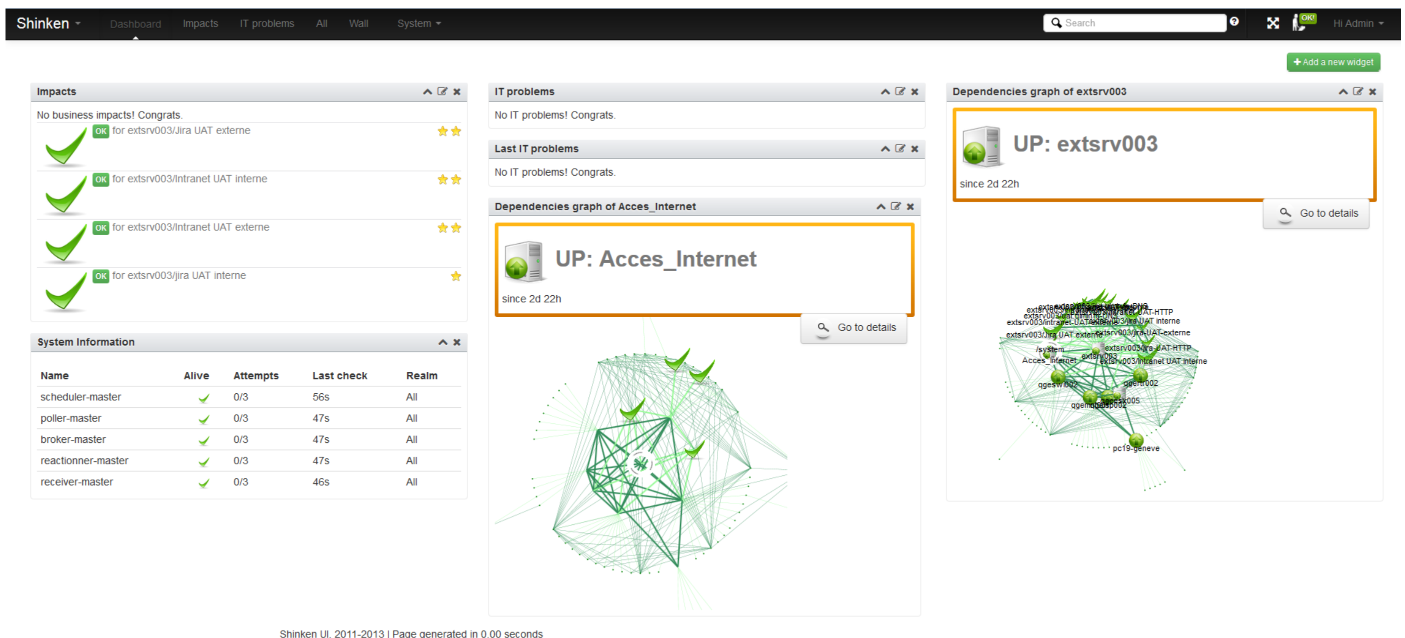
Un agent est un programme qui est déployé sur un système supervisé afin de fournir les données nécessaires au Poller. Exemple typique de ces agents ; nsclient qui s'installe sur machine Windows. Le déploiement d'un agent est une grande problématique car, pour une infrastructure assez hétérogène et grande le travail est complexe.
L'approche actuelle est une supervision sans agent. Mais supervision sans agent ne veut pas dire qu'il n'y rien d'installé sur la machine. C'est plutôt de superviser avec des programmes ou protocoles déjà installés. Dans la plupart de cas des services comme SSH, SNMP, WMI existent sur les machines.
Je privilégie aussi la supervision sans agent dans le but de faciliter le travail et faire moins d'installation possible sur l'infrastructure existante.
Mais une configuration est souvent nécessaire pour pouvoir faire supervision sans agent.

3.3.6. Interface Web de Shinken

Shinken vient avec sa propre interface web qui est assez moderne et ergonomique. L'interface de base est responsive et marche assez bien sur des appareils mobiles avec des vues plus simples. L'authentification de base est gérée avec un fichier et les préférences des utilisateurs sont sauvegardées dans une base de données simple comme sqLite ou mongoDB. L'authentification peut être couplée avec l'annuaire LDAP.



** Figure 15 Vues de l'interface web sur un smartphone*



** Figure 14 Interface Web Shinken - Vue Dashboard*

Les vues principales :

Dasboard : vue personnalisé pour chaque utilisateur.
All : affiche tous les éléments surveillés.
Impacts : affiches les règles métier.
IT problems : Affiches les problèmes, dans le cas de problèmes liés entre eux, elle affiche les sources des problèmes.
Wall : Impacts et les IT problems.
System : affiche les états des démons.
Currently : affiche une vue globale des éléments.

3.3.7. <u>*Outil de métrologie ou reporting*</u>

Shinken n'est pas une solution de supervision complète donc il faut ajouter des outils pour faire plus comme analyser les données de check (métrologie) ou faire des rapports de panne et disponibilité reporting. Le couplage avec d'autres outils est assez facile côté Shinken mais l'installation de ces outils est souvent problématique. Le cahier de charge nécessite un outil de reporting et l'outil de métrologie est un complément de cet outil. Mais en cours de route j'avais la possibilité de prendre l'initiative et discuter des possibilités à ajouter. La cartographie peut fournir un certain confort pour les administrateurs mais en même temps il peut aider à trouver la source physique d'un problème plus facilement.

a) Métrologie

Les outils de métrologie sont utiles pour analyser les performances d'un système informatique. Ces outils sont complémentaires aux de l'historique pour pouvoir comprendre tout ce qui se passe. Les outils historiques sont limités au niveau des détails et plutôt orientés vers les évènements, changements d'état ou log .Le couplage de ces deux outils crée la supervision. Car la supervision comprend aussi l'analyse de données.
Graphite et pnp4nagios sont plus utilisés et conseillés. Pnp4nagios a besoin du service npcd activé dans le shinken. J'ai pris Graphite car cela donne des graphiques assez compréhensibles avec des périodes mais aussi il récupère les données depuis Shinken avec le module Livestatus qui est disponible avec Shinken. Graphite a deux points vitaux.

- Sauvegarder les métriques time-series data.
- Faire des graphiques à partir de ces données.

Graphite est composé de trois parties

- *carbon* – le démon qui écoute pour les données temporelle.
- *whisper* – une librairie simple pour sauvegardes les données.
- *webapp* - a (Django) application web pour afficher les graphiques.



b) Reporting

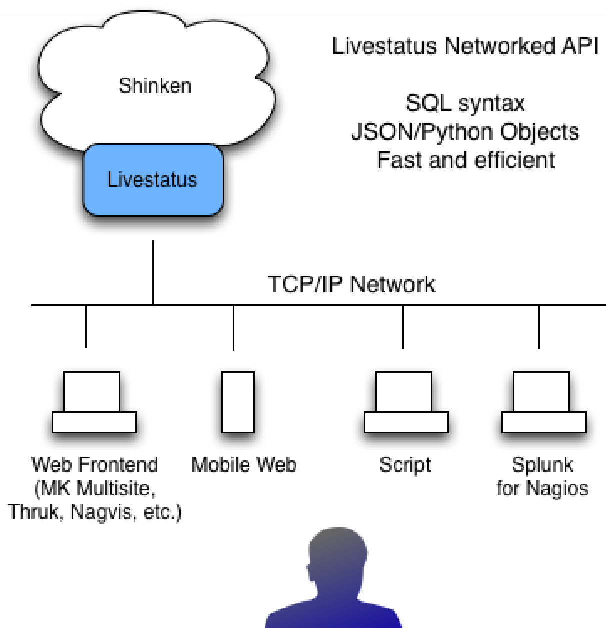
Thruk, Mk_multisite, Adagios sont des outils connus et fiables pour créer des rapports, surveiller les disponibilités du système informatique. Ces outils fournissent également des interfaces web complètes pour la supervision. On peut aussi regrouper plusieurs serveurs de supervision via ces outils.
J'ai installé multisite et thruk pour pouvoir comparer les possibilités et j'ai préféré le thruk car la gestion de reporting est plus avancée. Mais en même temps il est complèt au niveau de l'interface, l'entreprise peut désactiver l'interface Shinken et rester sur thruk pour monopoliser l'affichage. Le thruk permet aussi de regrouper plusieurs moteur de surveillance comme Shinken et Nagios. Dans le cas ou on doit déployer un autre serveur de surveillance chez OVH ou ailleurs le Thruk peut regrouper tous ces serveurs sur une seule interface. Il permet en même temps d'afficher les cartes crééer à partir de Nagvis.

c) Cartographie

Il y a aussi des outils pour afficher le système surveillé via des cartes géographique, des images ou des schémas créés manuellement. Le plus connu et utilisé, Nagvis, marche assez bien et je n'ai pas trouvé d'autre outils open source

** Figure 16 Interface Web avec Graphite*

3.3.8. Flux de données entre Shinken et les outils



Livestatus est le ré implémentation du module mk_livestatus de Nagios et il fait la brique de lien entre Shinken et d'autre outil comme Graphite, Thruk, Multisite etc. Module Livestatus communique en http sur le port 50000. Il est rapide et permet d'obtenir l'état des machines, services, la configuration de shinken, des rapports et statiques mais aussi envoyer des commandes vers Shinken via un langage de requête similaire à SQL mais performant. Il est unique module pour obtenir les impacts business et dépendances intelligents. Il peut marcher sans une base de données tout simplement avec un système de log.

** Figure 17 Schéma de communications entre shinken et d'autres outils*

Source : http://shinken.readthedocs.org/en/branch-1.4/89_packages/livestatus_shinken.html

3.3.9. Commun avec Nagios

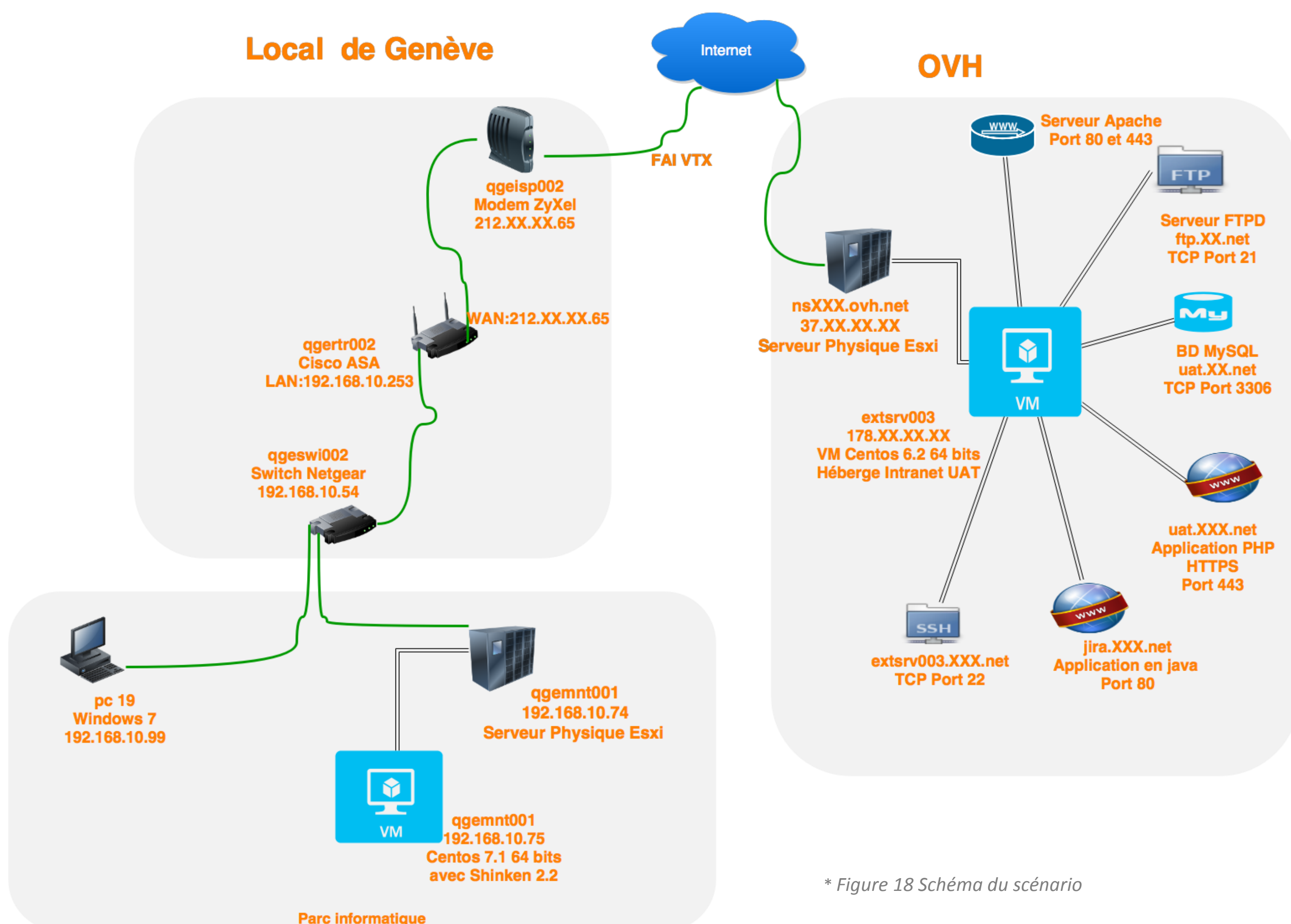
Les configurations sont assez similaires que Nagios et on peut garder les paramètres pour les éléments surveillés. On peut utiliser les mêmes plugins.

3.3.10. ***Corrélation de données avec Shinken***

Shinken a introduit la corrélation de données dans le monde supervision basé sur Nagios. Il est capable de déterminer la corrélation de base avec sa configuration standard. En plus de sa fonctionnalité de déterminer la source d'un problème qu'il appelle « root », il ajoute le terme dépendance et bp_rules. Il y a deux type de dépendances celui entre les hôtes et mais aussi un deuxième entre les services. Les règles métiers aident beaucoup pour faire la corrélation aussi. Avec activation « problème impacts states change », Shinken ne va pas alerter sans analyser les relations entre éléments à surveiller. Cela évite des alertes inutiles. Ex. : si un hôte tombe, on ne va pas alerter pour les services hébergés par cet hôte.

4. Scénario

4.1. Scénario Intranet



** Figure 18 Schéma du scénario*

Intranet Qim utilisé pour la gestion des différents éléments administratifs. L'intranet contient les fonctionnalités qui permettent de gérer :

- L'annuaire des collaborateurs
- Les comptes rendus d'activité
- Les demandes de congés
- Les notes de frais
- La facturation

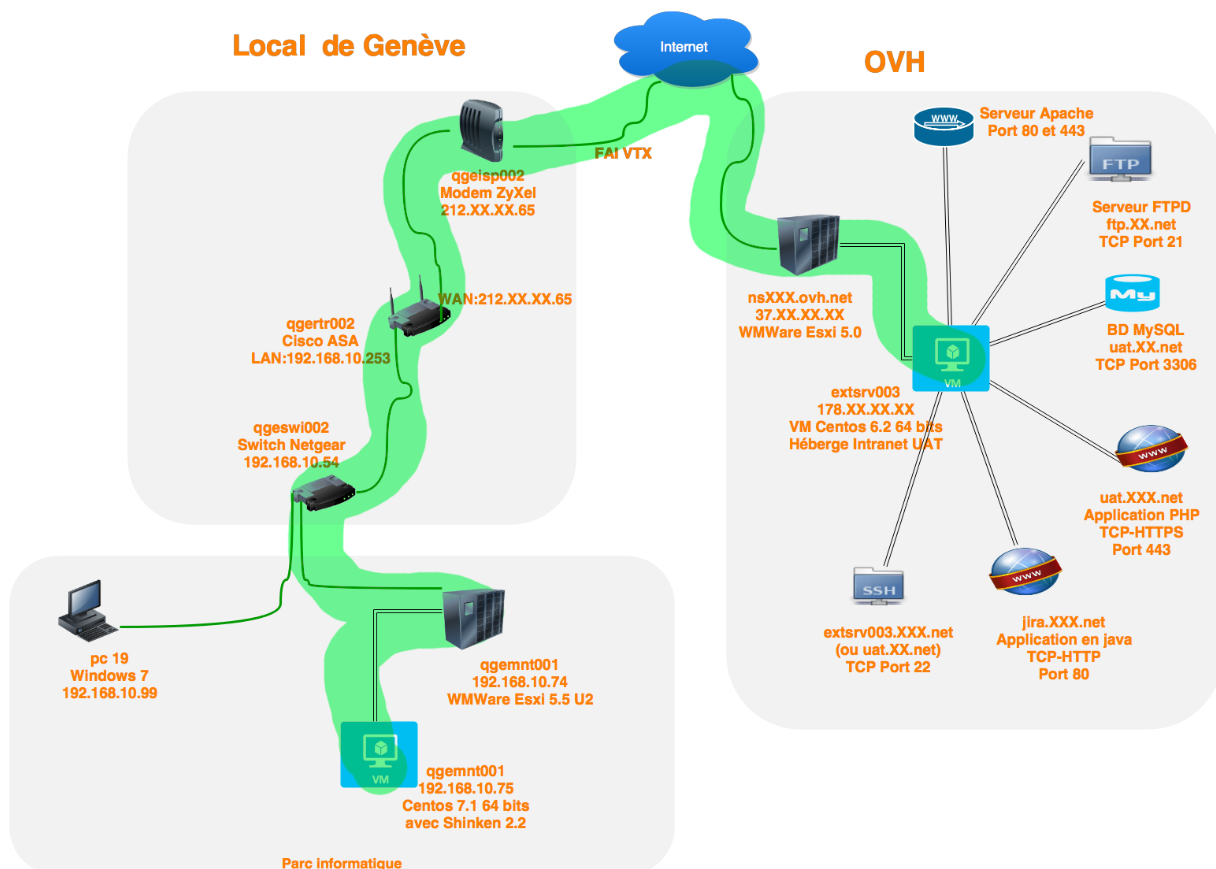
Serveur FTP utilisé pour permettre aux clients de déposer des fichiers directement sur le serveur public de Qim info hébergé par OVH.

Comme l'intranet est utilisé en permanence il sera important de superviser l'infrastructure qui est derrière. Ce scénario consiste à superviser une infrastructure intranet qui est basé sur un hyperviseur Esxi. Les employés utilisent intranet depuis le parc informatique de Genève. Pour ce scénario les éléments intéressant sont les dépendances car l'utilisation d'intranet passe par une chaîne depuis un pc jusqu'aux services.

Le schéma représente la connectivité depuis un pc de parc informatique dans le local de Genève vers l'intranet qui est hébergé chez OVH. Ce schéma aide à comprendre la chaîne de dépendances et il dévoile juste la partie concernant le scénario.

4.2. DÉPENDANCES

Les éléments physiques et les services à surveiller dépendent d'autres éléments de l'infrastructure ou service. Typiquement pour pourvoir accéder un site internet on doit passer par le switch qgeswi002, le routeur qgertr002, modem qgeesi002 donc l'accès internet dépend de tous ces équipements du réseau. Dans le cas d'une panne de cet équipement, les éléments dépendants seront touchés aussi. Cela veut dire que on ne peut avoir accès à internet sans avoir le switch qgeswi002, le routeur qgertr002 et le modem qgeesi002 qui marchent. Si l'un de ces équipements tombe en panne on ne peut pas savoir l'état des équipements qui sont derrière de cet équipement. Le serveur de supervision doit prendre en compte toutes ces dépendances pour gérer la source des pannes correctement. Le paramétrage de dépendances est aussi indispensable pour la corrélation de données. Si le routeur est en panne le serveur va pouvoir déterminer les équipements et services dépendants et il ne va pas faire des notifications pour ces éléments.



* *Figure 19 Schéma de dépendances*

4.3. MESURES DE SUPERVISION

	Que surveiller ?	Comment ?
Pour les Hôtes	**Charge CPU** : La charge moyenne de CPU.	Retourne le résultat de la commande mpstat sur l'hôte linux. Pour les hôtes Windows un check via WMI peut fournir les mêmes informations. Niveau warning :80%, niveau critical : 90%.
	RAM: Le taux de consommation de mémoire vive, le taux de consommation de mémoire Swap.	Retourne le résultat de la commande free –k sur l'hôte linux. Pour les hôtes Windows un check via WMI peut fournir les mêmes informations. Niveau warning :80%, niveau critical : 90%.
	Disque : L'espace disponible sur disque.	Retourne le résultat de la commande df –l –T –k -P sur le hôte linux. Pour les hôtes Windows un check via WMI peut fournir les mêmes informations. Niveau warning :75%, niveau critical : 90%.
	UP ou DOWN : Si le hôte allumé.	Ping (ICMP avec des paramètres) La commande peut être modifié si l'hôte refuse celle-là ou utiliser nping à la place. Ex :Icmp –H IP –w 3000,100% -c 5000,100% -p 10
Équipements réseaux (Switch, Routeur, Modem)	**Charge CPU** : La charge moyenne de CPU.	Un check en SSH retourne le résultat de la commande mpstat sur l'hôte. On peut obtenir même info via un check SNMP.Niveau warning :80%, niveau critical : 90%.
	UP ou DOWN : Si le hôte allumé.	Ping (ICMP avec des paramètres) La commande peut être modifié si le hôte refuse celle-là ou utiliser nping à la place.
	Paquets perdus	Via un check par Ping, on compte les paquets perdus. Mais aussi un check ssh peut fournir des informations de réseau.
	Bande passante	Via le log de mrtg ex : /var/lib/mrtg/192.168.1.253_1.log

Service, Application	**FTP** Tester le port 21 et le login ftp (ftp.xxx.net)	Check_ftp essaie de se connecter sur le serveur ftp avec le login d'un client. Un autre check tcp consiste à surveiller /proc et /sys avec une connexion SSH.
	MySQL (connexion serveur MySQL)	Le check MySQL se connecte au serveur avec un login pour tester la connexion.
	SSH connexion ssh	Le check ssh plugin va tester la connexion en essayant de se connecter au serveur SSH. Le plugin utilise la librairie pramiko pour se connecter en tant que client dans python.
	Serveur Apache Si le serveur tourne	Le plugin apache de nagios retourne le statu de serveur Apache. On peut aussi tester si le processus apache tourne avec un check ssh.
	Application métiers https://uat.xxx.net/ (Site Intranet en PHP) http://jira.xxx.net (Dashboard. Site versionning en java)	Tester les URI Tester les certificats Check l'entré DNS Avec le check http fournit par nagios. Le plugin se base sur des requêtes http afin de vérifier les applications. Mais il faut encore tester l'accès à ces application depuis réseaux externe de QIM Info soit en passant par un proxy, soit installant un Poller chez OVH ou encore en utilisant le plugin check my website.
FAI	**Accès internet** (Si on peut accéder internet depuis local Genève)	Ping simple vers un serveur dans l'internet.

** Tableau 2 Détails de mesures de la supervision*

4.4. BUSINESS ACTIVITÉ MONITORING

Le concept de business activité monitoring (BAM) comprend l'acquisition, l'agrégation, l'analyse et la présentation en temps réel de données (typiquement des séquences de valeurs temporelles et leur évolution) associées à des processus d'entreprise. Ces données sont souvent obtenues dans le contexte d'un processus d'entreprise modélisé par des activités amont de modélisation de procédure d'entreprise (BPM). Cependant, le BAM peut être employé indépendamment de l'existence d'une solution de BPM. Le BAM consiste en une solution d'entreprise destinée à fournir en temps réel un résumé de la situation des activités métiers aux responsables des opérations et la direction.

* Source :http://fr.wikipedia.org/wiki/Business_activity_monitoring

Le but d'une solution BAM est, entre autres, de permettre une réaction au plus tôt grâce à un système d'alarmes en cas de dérive et, dans le meilleur des cas, pouvoir agir de manière proactive.
On utilisera une approche TOP-DOWN (du métier vers la technique) pour mettre en place une solution de ce type.

4.5. RÈGLES MÉTIER (BUSINESS RULES)

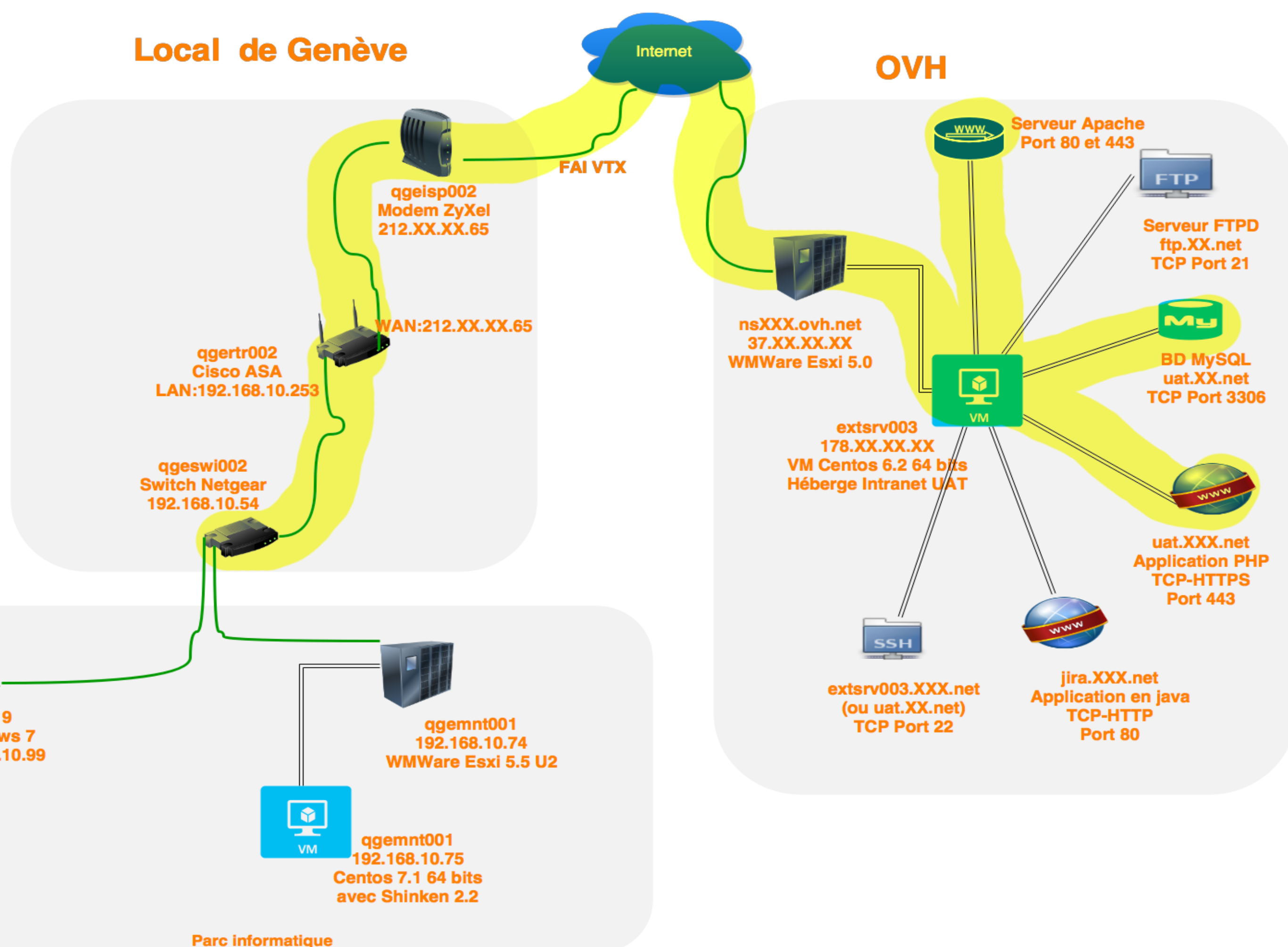
Règles business et notion d'importance business sont des points forts de Shinken par rapport à Nagios. Cela permet aux administrateurs de se concentrer sur ce qui est « important ». Surveillance IT n'est plus suffisant, il faut savoir surveiller ceux qui ont un impact important sur le business de l'entreprise.
Pour cela il faut un service supplémentaire dans la supervision pour agréger certains éléments.
Parmi les éléments à surveiller deux applications métiers (ERP) ont une importance pour le business de l'entreprise.

4.5.1. *Définir des règles métiers (Business rule)*

On peut ajouter une règle métier comme un host ou service. La seule différence sera mettre bp_rule comme commande en précisant les éléments de la règle. Il ne faut pas implémenter cette commande car elle est interne à Shinken. On peut combiner plusieurs éléments pour définir une règle avec des expressions régulières comme et, ou, non etc.
J'ai défini deux règles pour l'intranet car il y a deux accès diffèrent qui impliquent l'utilisation d'éléments différents ; la première est pour l'accès interne (depuis local Qim Info de Genève) et la deuxième est pour l'accès depuis internet (pour les collaborateurs externes).
L'application intranet est utilisée depuis local de Genève mais aussi depuis internet pour collaborateurs.

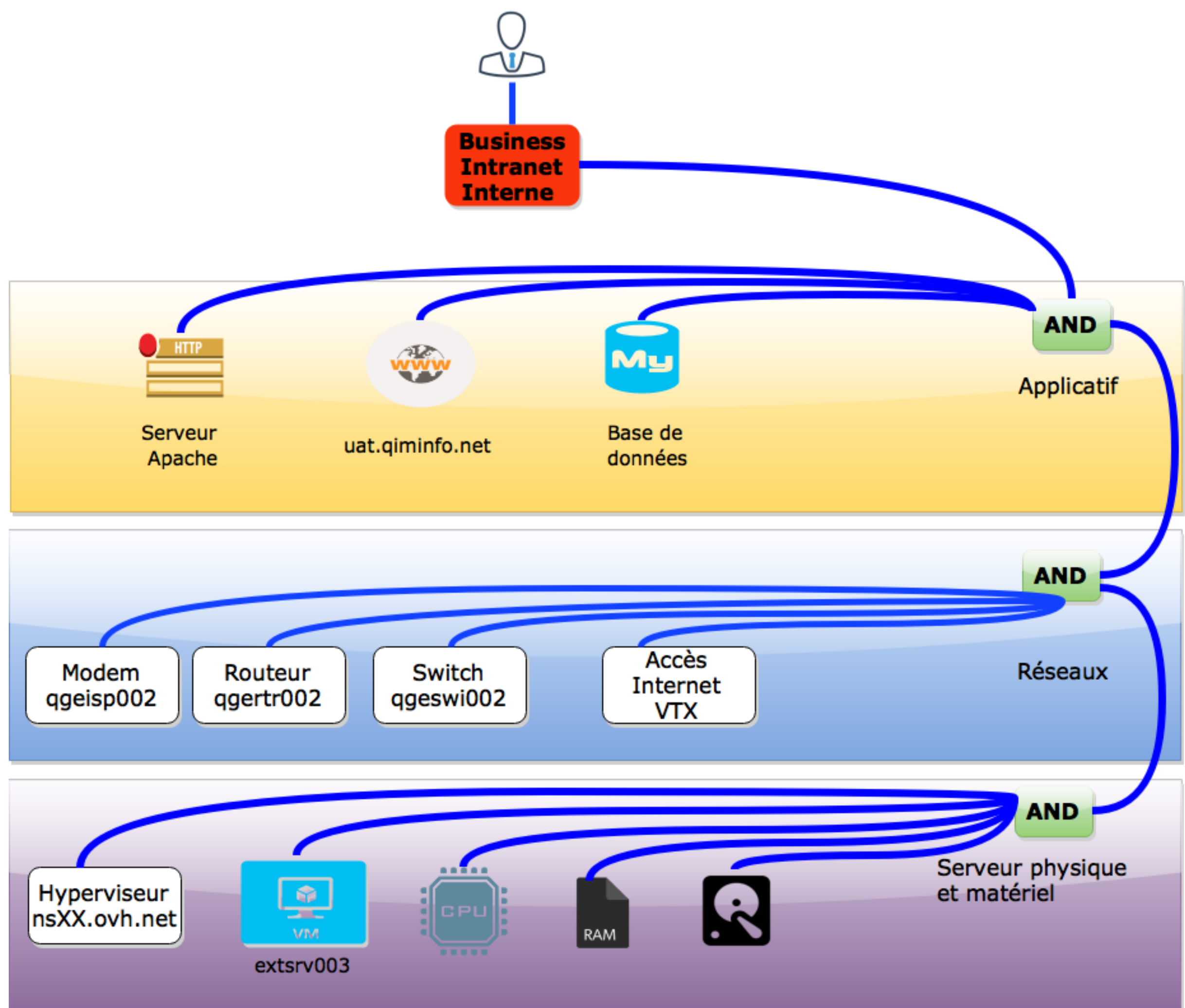
Intranet interne :
URI&mysql & http_serveur & VM & machine_Esxi & acces_internet & routeur & modem &switch
Intranet_externe :
URI &mysql & http_serveur & VM & machine_Esxi & ping_externe
intranet_Dashboard:
URI &mysql & http_serveur & VM & machine_Esxi & acces_internet & routeur & modem &switch

& : Et exemple ; site_web=mysql& http_serveur on doit avoir mysql et serveur http opérationnel pour que le business site_web tourne.
| : Ou ; site_web1=(mysql1 | mysql2) & (http_serveur) le business site_web1 a besoin l'un des serveurs mysql1 ou mysql2 et le serveur http en même temps



** Figure 20 Schéma Régler métier (Business Rule) utilisation intranet depuis local de Genève*

Une règle métier explique ce qu'il faut pour faire tourner le business avec certaines expressions régulières comme la figure-21 ci-dessous.

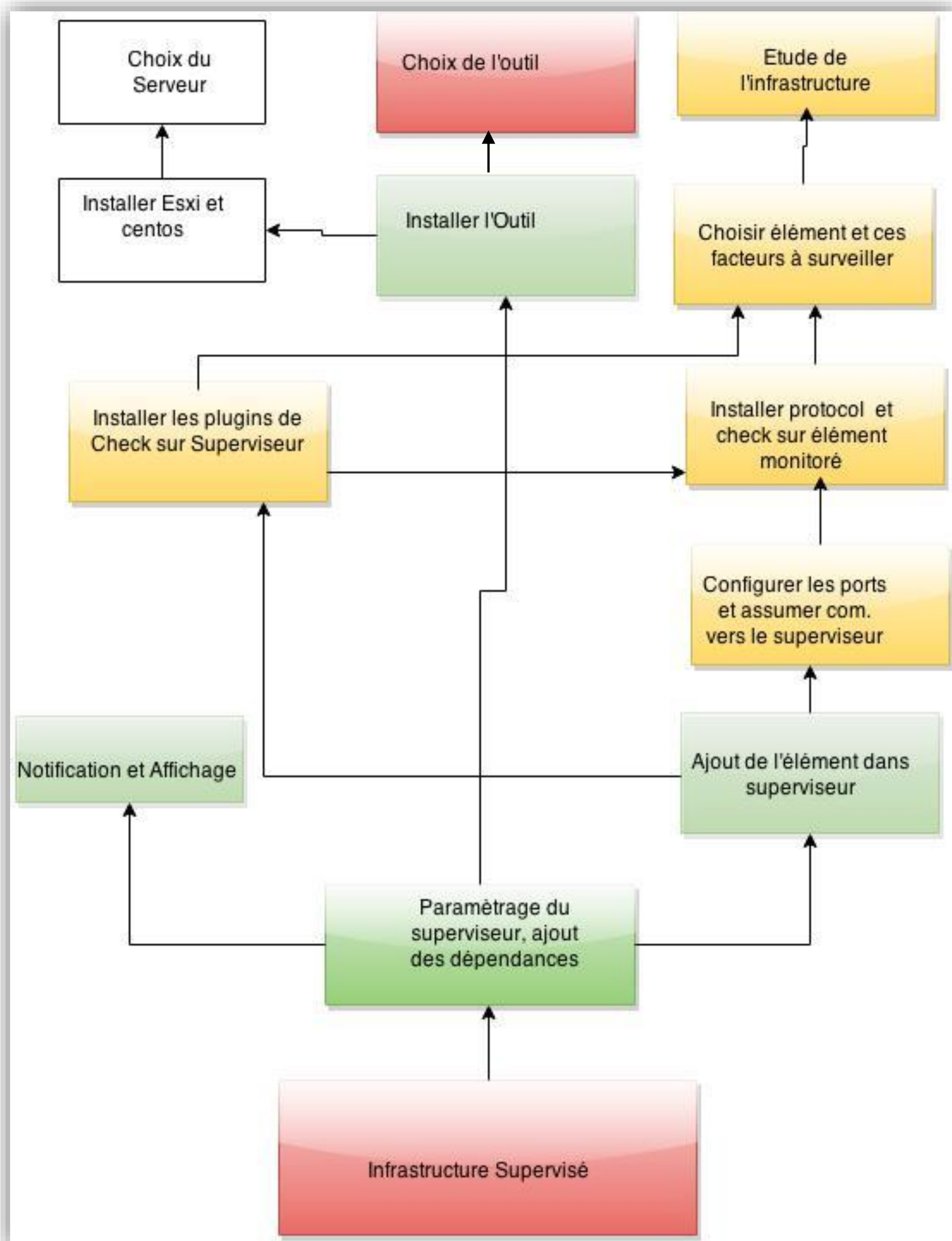


* *Figure 21 Composition d'un Règles métier (Business rule)*

4.6. STRATÉGIE DE DÉPLOIEMENT

L'outil de supervision s'installe sur un serveur dans le local de Genève.

Le logigramme ci-dessous montre les étapes de réalisations pour la supervision mais elle donne en même temps une vision globale du projet et aide à la décomposition du projet.



* Figure 22 Logigramme de déploiement

5. Mise en œuvre

La partie de mi sen en œuvre comprend la réalisation des tâches définies dans le cahier de charges (section 2.7)

5.1. Choix de l'architecture et du système d'exploitation

L'entreprise possède déjà des hyperviseurs Esxi (Esxi 5.0 et Esxi 5.5) et des machines virtuelles avec Centos (version 6.2 et 6.3) comme système d'exploitation. Centos est utilisé aussi par les développeurs de Shinken. Donc le choix de l'architecture se porte sur une machine virtuelle basé Hyperviseur Esxi 5.5 avec Centos 7.1 par l'intérêt de l'entreprise. La machine virtuelle peut facilement être exportée et déployée sur un autre hyperviseur dans le local.

5.2. Les Outils Réseaux

Certains outils réseaux peuvent être utiles pour la surveillance afin de comprendre ce qui se passe dans le réseau. J'ai choisi les indispensables et j'ai installé nmap et Wireshark n'étaient pas déjà installé avec Centos.

- nmap
- tcpdump
- nping
- netcat
- wireshark

Avec ces outils on peut :

- Détecter les ports ouverts
- Détecter les équipements réseaux
- Détecter si un nœud de réseau est atteignable

Installation namp et Wireshark :

[hbi@qgemnt001]#sudo yum install nmap
[hbi@qgemnt001]# sudo yum install wireshark

5.1. Prérequis

Il faut installer les dépendances comme pip, pycurl et cherrypy3. Le Shinken a besoin au moins le Python 2.6 mais 2.7 est conseillé. Python 3.0 n'est pas encore tout a fait compatible avec Shinken.
La plupart des OS vient avec Python installé. La version 7.1 de Centos inclut la version 2.7 de Python.
[hbi@qgemnt001]# sudo install python-pip python-pycurl python-cherrypy3
[hbi@qgemnt001]# sudo useradd --user-group shinken

5.2. ACTIVER OU DÉSACTIVER SELINUX

SELinux est le système disponible sous des distributions récentes de Linux permettant de définir une politique de sécurité d'accès très fine par rapport au système d'exploitation. Il se positionne en plus des classiques Firewall (permettant seulement un filtrage au niveau réseau). Dans notre cas SELinux peut perturber la supervision car certains modules ne marchent pas avec SELinux activé.

Pour le désactiver, deux solutions: La première est d'éditer le fichier */etc/selinux/config* et de remplacer la ligne **SELINUX=enforcing** par **SELINUX=disabled** puis de rebooter le système.
La seconde, plus rapide et bien utile pour faire des tests est de taper en ligne de commande (root):
[hbi@qgemnt001]# sudo /usr/sbin/setenforce 0 -> ici, pas de reboot nécessaire.

5.1. INSTALLATION SHINKEN

Il faut toujours continuer à installer avec même méthode d'installation sinon il faut tout désinstaller et réinstaller (PIP dans notre cas). Il faut se connecter en tant que root mais les commandes sont lancées avec shinken via utilisateur shinken. Avec la commande pip on installe la dernière version. Actuellement la dernière version est 2.4 RC mais comme il vient de sortir, j'ai installé la version 2.4. Pour savoir la version de Shinken utilisé il faut exécuter
Les installation deivent être faites en tant que root.
[hbi@qgemnt001]# sudo pip install shinken
[hbi@qgemnt001]# sudo shinken --init

A partir de la version 2, Shinken utilise LSB pour la hiérarchie de ces répertoires.

- */etc/shinken* for fichier de configurations
- */var/lib/shinken* for shinken modules
- */var/log/shinken* for fichiers de log
- */var/run/shinken* for fichiers de pid

Après l'installation la première chose est à faire est d'installer et configurer l'interface web.L'utilisation de l'interface web nécessite un module pour gérer l'authentification et un module de base de données pour sauvegarder les préférences des utilisateurs. Le module auth-cfg-password gère l'authentification via les fichiers de contact. Pour la sauvegarde on peut utiliser mongodb ou sqlitedb.

J'ai testé les deux modules et j'ai choisi sqlitedb, dans le cas de mongodb il faut installer mongod sur la machine.
[shinken@qgemnt001] # shinken install webui
[shinken@qgemnt001] # shinken install auth-cfg-password
[shinken@qgemnt001] # shinken install sqlitedb (OU **shinken install mod-mongodb** pour une bd NOSQL)
*Dans le cas de mongodb le serveur mongod doit être installé sur le serveur en suivant les instructions :
http://docs.mongodb.org/manual/tutorial/install-mongodb-on-red-hat/

Pour modifier le mot de passe par défaut:
[shinken@qgemnt001] # vi /etc/shinken/contacts/admin.cfg
Modifier le champ password, on peut aussi modifier le nom avec contact_name.

Les modules d'authentification et sauvegarde sont ajoutés dans modules webui.
[shinken @qgemnt001] # vi /etc/shinken/modules/webui.cfg
Et ajouter les modules :
Modules auth-cfg-password, sqlitedb
Le module webui est lancé par le broker donc il faut ajouter webui comme module dans la configuration du broker.
[shinken @qgemnt001] # vi /etc/shinken/broker/broker-master.cfg
Et ajouter webui

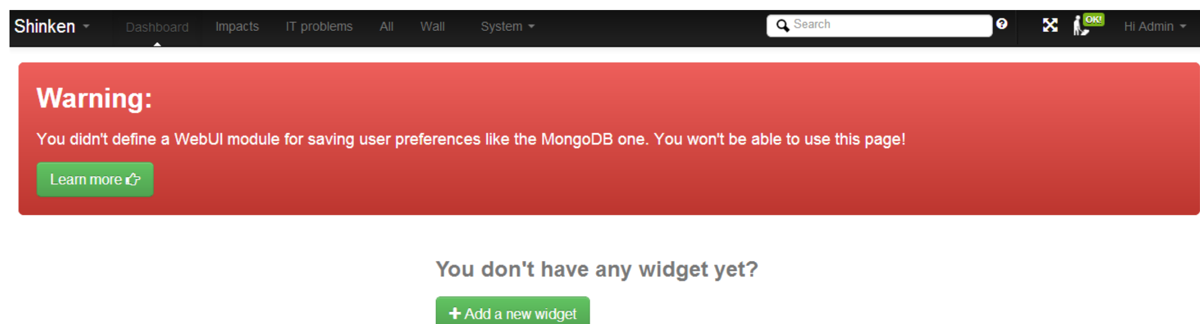
modules webui

Démarrage automatique de Shinken au démarrage d'ordinateur :

[root@qgemnt001]# chkconfig shinken on

Démarrage de shinken manuellement :
[shinken@qgemnt001]# /etc/init.d/shinken start
Option –d permet de démarrage en mode debugg.
[shinken @qgemnt001]# /etc/init.d/shinken –d start

Ou encore les commandes suivantes :
[shinken @qgemnt001]# service shinken start
[shinken @qgemnt001]# systemctl start shinken
Dans le cas où le module d'authentification n'est pas chargé correctement interface web refusera le login. Un autre problème fréquent est l'activation du module de sauvegarde. L'interface web affiche une erreur comme ci-dessous en cas de problème avec ce module.



** Figure 23 Message d'erreur si le module de sauvegarde n'est pas activé.*

Après le redémarrage de Shinken la première erreur arrive car la page Wall de l'interface web ne s'affiche pas.
Message d'erreur : Error 404: Not Found
Sorry, the requested URL 'http://localhost:7767/wall' caused an error:
Ce problème est connu est la solution est :
[root@qgemnt001#] vi /var/lib/shinken/modules/webui/plugins/wall/wall.py
Remplacer la ligne 31 import.helper par import shinken.misc.sorter et recompiler le fichier.

Les processus de Shinken sont exécutés avec l'utilisateur shinken en tant que des processus Python avec des fichiers de configurations précisés. Le nombre de processus subit des changements par rapport à la configuration. La figure 24, ci-dessous, montre le nombre des processus après tous les ajouts de services et les machines qui ont augmenté par rapport au nombre de processus de début.

```
[root@qgemnt002 modules]# ps -Af | grep shinken
shinken   2360     1  1 08:38 ?        00:00:54 python2.7 /usr/bin/shinken-scheduler -d -c /etc/shinken/daemon
s/schedulerd.ini
shinken   2363  2360  0 08:38 ?        00:00:00 python2.7 /usr/bin/shinken-scheduler -d -c /etc/shinken/daemon
s/schedulerd.ini
shinken   2468     1  0 08:38 ?        00:00:33 python2.7 /usr/bin/shinken-poller -d -c /etc/shinken/daemons/p
ollerd.ini
shinken   2477  2468  0 08:38 ?        00:00:01 python2.7 /usr/bin/shinken-poller -d -c /etc/shinken/daemons/p
ollerd.ini
shinken   2531     1  0 08:38 ?        00:00:33 python2.7 /usr/bin/shinken-reactionner -d -c /etc/shinken/daem
ons/reactionnerd.ini
shinken   2538  2531  0 08:38 ?        00:00:01 python2.7 /usr/bin/shinken-reactionner -d -c /etc/shinken/daem
ons/reactionnerd.ini
shinken   2557     1  1 08:38 ?        00:00:42 python2.7 /usr/bin/shinken-broker -d -c /etc/shinken/daemons/b
rokerd.ini
shinken   2560  2557  0 08:38 ?        00:00:01 python2.7 /usr/bin/shinken-broker -d -c /etc/shinken/daemons/b
rokerd.ini
shinken   2582     1  0 08:38 ?        00:00:32 python2.7 /usr/bin/shinken-receiver -d -c /etc/shinken/daemons
/receiverd.ini
shinken   2588  2582  0 08:38 ?        00:00:00 python2.7 /usr/bin/shinken-receiver -d -c /etc/shinken/daemons
/receiverd.ini
shinken   2604     1  0 08:39 ?        00:00:27 python2.7 /usr/bin/shinken-arbiter -d -c /etc/shinken/shinken.
cfg
shinken   2605  2604  0 08:39 ?        00:00:00 python2.7 /usr/bin/shinken-arbiter -d -c /etc/shinken/shinken.
cfg
shinken   2665  2468  0 08:39 ?        00:00:00 python2.7 /usr/bin/shinken-poller -d -c /etc/shinken/daemons/p
ollerd.ini
shinken   2695  2557  0 08:39 ?        00:00:01 python2.7 /usr/bin/shinken-broker -d -c /etc/shinken/daemons/b
rokerd.ini
shinken   2718  2468  0 08:39 ?        00:00:00 python2.7 /usr/bin/shinken-poller -d -c /etc/shinken/daemons/p
ollerd.ini
shinken   2731  2531  0 08:39 ?        00:00:00 python2.7 /usr/bin/shinken-reactionner -d -c /etc/shinken/daem
ons/reactionnerd.ini
hbi      26363     1  0 09:15 ?        00:00:01 gedit /etc/shinken/modules/webui.cfg
root     28529  4428  0 09:38 pts/0    00:00:00 grep --color=auto shinken
```

** Figure 24 Listes de processus de Shinken après le démarrage*

5.1.1. Installation des modules, packs et plugins pour Shinken

Les packs contiennent les fichiers de configuration nécessaires pour ajouter les éléments et ces facteurs dans la supervision. Un fichier de pack contient la définition d'un host, ces paramètres, les services et commandes à utilisés. Ils ont conçu pour que certains éléments puissent être supervisés plus facilement. Ils sont comme des templates et on les utilise juste mettant leurs nom dans la configuration d'un host ou service comme ci-dessous :

use linux-snmp

L'installation des packs ou modules se fait de la même manière.
On peut chercher les packs ou des modules via la commande
[shinken @qgemnt001#] shinken search nom_pack_ou_module
J'ai installé les packs ci-dessous après les recherches avec la commande :
[shinken @qgemnt001] # shinken install nom_pack_ou_module
Mais on peut aussi trouver ces packs ou modules sur le site de Shinken : http://shinken.io

Pour afficher tous les paquets disponibles :
[shinken @qgemnt001] # shinken search all
Pour avoir une interface web qui marche-t-il faut installer les modules webui, auth-conf pour le login et un module de base de données pour les préférences des utilisateurs comme mod-mongodb ou sqlitedb.
Afficher les paquets que vous avez installés
[shinken @qgemnt001] # shinken inventory
Vue la diversité de la couche matériel et applicative infrastructure de l'entreprise j'ai essayé d'installer tous les packs qui peuvent être utiles.

- Sqlitedb -> base de données pour webui
- logstore-sqlite -> pour les logs
- linux-ssh -> pour le check de machines linux en SSH
- webui - > pour avoir l'interface web native de Shinken
- mongodb,mod-mongodb -> module de mondodb pour sauvegarde de données
- auth-cfg-password -> module d'authentification

- wmware -> check machine WMWare
- check_mywebsite -> pack pour le plugin check mywebsite
- http -> pack pour check en http
- iis (w server) -> pack pour check d'un serveur Windows
- windows (w checks) -> pack pour check machine Windows
- linux snmp -> pack pour le check en snmp
- router -> pack pour check de routeur
- switch -> pack pour le check de switch
- mysql -> pack pour le check de serveur base de données
- ftp -> pack pour le check de service ftp
- cisco -> pack pour le check des équipements Cisco
- dhcp -> pack pour le check service DHCP
- dns -> pack pour le check service Dns
- cups (imprimantes) -> pack pour les check d'imprimante

Cette configuration nous permet d'ajouter des services, hosts mais pour faire des check il faut aussi installer des plugins (exécutables). Shinken et d'autre outils de supervision comme Icinga, Nagios, Semsu ont des check communs regroupé dans un package nommé monitoring plugins sous le site :
https://www.monitoring-plugins.org/ .
Télécharger les plugins.

[hbi@qgemnt001]# sudo wget --no-check-certificate https://www.monitoring-plugins.org/download/monitoring-plugins-2.1.1.tar.gz

[hbi@qgemnt001]# sudo tar -xvf monitoring-plugins-2.1.1.tar.gz
[hbi@qgemnt001]# sudo cd monitoring-plugins-2.1.1
Configurer et installer les plugins suite à l'installation du compilateur GCC.
[hbi@qgemnt001]# sudo yum –enabledrepo=rpmforge intall gcc
[hbi@qgemnt001]# sudo ./configure
(Où configuration optimale pour Shinken
[root@qgemnt001]# ./configure --with-nagios-user=shinken --with-nagios-group=shinken --enable-libtap --enable-extra-opts --enable-perl-modules --libexecdir=/var/lib/shinken/libexec)
[hbi@qgemnt001]# sudo make install

On peut installer aussi les plugins faits pour Nagios car il y a enormoments de plugins pou Nagios.
[hbi@qgemnt001]# sudo install nagios-plugins
Le site https://exchange.nagios.org/directory/Plugins nous aide à trouver des plugins presque pour tout et l'utilisation est aussi expliquée sur le site.
Après installations de ces plugins il faut tester chaque plugin en commande ligne avant d'utilisation. Pour les plugins téléchargés depuis le monitoring-plugin.org une page de manuel existe pour leurs utilisations.
Page de manuel pour le monitoring plugins : https://www.monitoring-plugins.org/doc/man/index.html
Tous les plugins donnent aussi son utilisation avec l'option –h ou –help.

```
[root@qgemnt001 libexec]# ./check_tcp_states_by_ssh.py --help
Usage: check_tcp_states_by_ssh.py [options]

Options:
  --version             show program's version number and exit
  -h, --help            show this help message and exit
  -H HOSTNAME, --hostname=HOSTNAME
                        Hostname to connect to
  -p PORT, --port=PORT  SSH port to connect to. Default : 22
  -i SSH_KEY_FILE, --ssh-key=SSH_KEY_FILE
                        SSH key file to use. By default will take
                        ~/.ssh/id_rsa.
  -u USER, --user=USER  remote use to use. By default shinken.
  -P PASSPHRASE, --passphrase=PASSPHRASE
                        SSH key passphrase. By default will use void
```

* *Figure 25 Aide pour l'utilisation d'un plugin*

Les plugins de Shinken se trouvent dans */var/lib/shinken/libexec* et les plugins de Nagios se trouvent */usr/lib/Nagios/*Plugins mais par rapport à la version téléchargé les plugins de Nagios peuvent se trouver aussi dans */usr/local/Nagios/libexec.* Le répertoire d'installation n'a pas de grande importance car dans les configurations de Shinken on définit les répertoires des plugins dans le fichier */etc/shinken/ressource.d/paths.cfg* et dans chaque commande on donner le chemin absolu d'un plugin avec ces macros.

```
# Nagios legacy macros
$USER1$=$PLUGINSDIR$
$NAGIOSPLUGINSDIR$=/usr/lib/nagios/plugins

#-- Location of the plugins for Shinken
$PLUGINSDIR$=/var/lib/shinken/libexec
```

* *Figure 26 Figure Configuration de chemins de plugins.*

5.1.2. Paramétrage de Shinken

La configuration de Shinken se trouve dans le répertoire */etc/shinken* et à chaque démarrage Shinken lit tous les fichiers qui finissent avec l'extension « cfg » qui se trouvent dans ce répertoire ou dans les sous-répertoires de celui-ci. Le fichier principal de la configuration est « */etc/shinken/shinken.cfg »*. Après chaque changement de la configuration il faut d'abord tester si la configuration est acceptée par Shinken via la commande ci-dessous.

[shinken @qgemnt001]# /usr/bin/shinken-arbiter –v –c /etc/shinken/shinken.cfg

J'ai ajouté un raccourci pour cette commande afin de faciliter la tâche.

[shinken @qgemnt001]# vim ~/.bashrc

Ajoute la ligne ;

alias chkShinken="/usr/bin/shinken-arbiter –v –c /etc/shinken/shinken.cfg"

Cette commande vérifie toute la configuration et indique un message d'erreur pour identifier le problème. Dans certains cas cela peut tromper car si on met un nom de template faux il ne va pas détecter ça comme erreur. Je l'ai constaté pendant les tests. Ayant mal écrit le generic-host pour un host shinken n'as indiqué aucune erreur mais le host est considéré UP car en cas de manque de check un service ou host est toujours Ok.

```
[1436281895] INFO: [Shinken] Checking reactionners...
[1436281895] INFO: [Shinken]    Checked 1 reactionners
[1436281895] INFO: [Shinken] Checking pollers...
[1436281895] INFO: [Shinken]    Checked 1 pollers
[1436281895] INFO: [Shinken] Checking brokers...
[1436281895] ERROR: [Shinken] [item::broker-master] Error: the module ddd is unknown for broker-master
[1436281895] ERROR: [Shinken] [item::broker-master] Error: the module ddd is unknown for broker-master
[1436281895] ERROR: [Shinken] [items] In broker-master is incorrect ; from /etc/shinken/brokers/broker-master.cfg:15
[1436281895] ERROR: [Shinken]    brokers conf incorrect!!
[1436281895] INFO: [Shinken]    Checked 1 brokers
[1436281895] INFO: [Shinken] Checking receivers...
[1436281895] INFO: [Shinken]    Checked 1 receivers
[1436281895] INFO: [Shinken] Checking resultmodulations...
[1436281895] INFO: [Shinken]    Checked 0 resultmodulations
[1436281895] INFO: [Shinken] Checking discoveryrules...
[1436281895] INFO: [Shinken]    Checked 24 discoveryrules
[1436281895] INFO: [Shinken] Checking discoveryruns...
[1436281895] INFO: [Shinken]    Checked 1 discoveryruns
[1436281895] INFO: [Shinken] Checking businessimpactmodulations...
[1436281895] INFO: [Shinken]    Checked 0 businessimpactmodulations
[1436281895] INFO: [Shinken] Cutting the hosts and services into parts
[1436281895] INFO: [Shinken] Creating packs for realms
[1436281895] INFO: [Shinken] Number of hosts in the realm All: 10 (distributed in 1 linked packs)
[1436281895] INFO: [Shinken] Total number of hosts : 10
[1436281895] ERROR: [Shinken] Configuration is incorrect, sorry, I bail out
Configuration is incorrect, sorry, I bail out
```

**Figure 27 Message d'erreur après une vérification de la configuration.*

La configuration de Shinken détaillée est expliquée sur site officiel de la documentation Shinken :
http://shinken.readthedocs.org/en/latest/08_configobjects/index.html
Je vais donner quelques exemples de configuration que j'ai faite.

a) Ajouter un host (machine, hôte)

Le répertoire /etc/shinken/hosts est prévu pour les hôtes. Pour faciliter le paramétrage chaque host déclaré est dans un fichier à part avec le nommage nom_mahcine.cfg. Les noms de fichiers ou hosts sont définies avec la convention de nommage de la Qim info. Pour la déclaration certaines champs sont obligatoires comme :

- host_name : nom de la machine
- address : IP ou FQDN de la machine
- max_check_attempts : nombre d'essai de check en cas de problème
- contacts : contact pour les notifications
- contacts_groups : groupe de contact pour les notifications
- notification_interval : l'intervalle de notification en cas de problème

Il y a d'autre champs utiles mais pas obligatoires comme :

- **parents** : la relation entre les machines, cela permet au Shinken de déterminer si un hôte est down ou unreachable(inaccessible), ce champ est très utile pour trouver la source dans le cas plusieurs problème si cela vient d'un élément perturbateur. On peut même annuler les notifications pour les hôtes unreachable avec les options de notifications.
- use : le Template à utiliser
- hostgroups : étiquette pour regrouper les machines
- check_commande : commande à exécuter pour tester la machine

- check_period : le nom de la période pour exécuter le check
- notification_enabled : si la notification est activée
- business_impact : l'impact de la machine sur business
- icon_set : l'icône à afficher pour la machine

Au lieu de configurer toutes les paramètres on peut créer des templates ou utiliser les templates existants comme generic-host. Dans la plupart de cas j'ai utilisé generic-host qui teste la disponibilité d'une machine avec un Ping. Même si on utilise un template pour une machine on peut toujours modifier les configurations en surchargeant le mêmes variables dans le fichier de la configuration.

```
define host{
        use                     generic-host
        alias                   Routeur
        contact_groups          admins
        host_name               qgertr002
        address                 192.168.10.253
        labels                  network
        hostgroups              geneve
        check_interval          5
        retry_interval          1
        parents                 qgeswi002,qgeesx005,qgemnt001

    _SNMPCOMMUNITYREAD  [redacted]
    _SNMP_MSG_MAX_SIZE  65535

    _LOAD_WARN          2,2,2
    _LOAD_CRIT          3,3,3
    _STORAGE_WARN       90
    _STORAGE_CRIT       95
    _CPU_WARN           80
    _CPU_CRIT           90
    _MEMORY_WARN        80
    _MEMORY_CRIT        95
    _NTP_WARN           0.128
    _NTP_CRIT           1
    _NET_IFACES         eth\d+|em\d+
    _NET_WARN           90,90,0,0,0,0
    _NET_CRIT           0,0,0,0,0,0

    _CHKLOG_CONF        $PLUGINSDIR$/logFiles_linux.conf
    _STORAGE_PATH       /


        }
```

** Figure 28 Définition d'un routeur via SNMP*

Le protocole ICMP est bloqué au niveau du firewall du routeur (qgertr002) donc pour contourner le problème je vérifie l'accessibilité du serveur extsrv003 via un port avec une requête TCP. J'ai créé un autre template à partir du generic-host sans la commande Ping pour cela.

```
define host{
        use                     generic-host-sans-check
        alias                   Intranet-QIM
        contact_groups          admins
        host_name               extsrv003
        address                 178.[redacted]
        hostgroups              ovh
        labels                  ovh
        icon_set                server
        parents                 Acces_Internet,qgeisp002,qge
        check_command           check_tcp !22
   _SSH_KEY                     $SSH_KEY$
   _SSH_KEY_PASSPHRASE          $SSH_KEY_PASSPHRASE$
   _SSH_USER                    $SSH_USER$
   _SSH_PORT                    $SSH_PORT$

   _LOAD_WARN                   1,1,1
   _LOAD_CRIT                   2,2,2
   _STORAGE_WARN                90
   _STORAGE_CRIT                95
   _STORAGE_UNIT                GB
   _STORAGE_MOUNTS              /
   _CPU_WARN                    80
   _CPU_CRIT                    90
   _MEMORY_WARN                 85
   _MEMORY_CRIT                 95
}
```

** Figure 29 Définition d'un serveur et configuration SSH*

b) Ajouter un service

Un service est lié à un hôte et il doit être déclaré via un fichier situé dans */etc/shinken/*services. Un service fait appel à une commande qui se trouve dans */etc/shinken/commandes.* À part les hôtes tous les éléments surveillés sont déclarés en tant qu'un service. Chaque service est lié à un hôte via host_name. Les champs ci-dessous sont obligatoires. En utilisant des template on déclare le minimum nécessaire et les autres paramètres sont pris en compte depuis les templates.

- host_name : le nom de l'hôte dont le service dépend
- service_description : le template du service à utiliser
- check_command : nom de la commande à exécuter
- max_check_attempts : nombre d'essai maximum de check en cas du résultat non OK
- check_interval : l'intervalle des checks
- retry_interval : l'intervalle de recheck en cas du résultat non OK
- check_period : période de lancement de check
- notification_interval : durée de notification en cas de changement d'état
- contacts : contacte pour les notifications
- contact_groups : groupe de contact pour les notifications

```
define service{
   service_description    SSH Connexion
   use                    linux-ssh-service
   host_name              extsrv003
   check_command          check_ssh_connexion
   _SSH_KEY                     $SSH_KEY$
   _SSH_KEY_PASSPHRASE          $SSH_KEY_PASSPHRASE$
   _SSH_USER                    $SSH_USER$
   _SSH_PORT                    $SSH_PORT$
}
```

* *Figure 30 Définition d'un service pour connexion SSH*

c) Ajouter un contact

Un contact est utilisé pour les notifications et pour l'authentification de l'interface web en cas de l'utilisation du module auth-cfg-password. En utilisant un template on doit juste définir le nom, l'adresse mail, mot de passe et les rôles. Les fichiers de contacts se trouvent dans */etc/shinken/contacts.*

```
define contact{
   use              generic-contact
   contact_name     admin
   email            xxxx@gmail.com
   pager            0600000000
   password         xxxx
   is_admin         1
   expert           1
}
```

* *Figure 31 Ajout d'un contact*

d) Ajouter un groupe de contact

Un groupe de contact permet de faire des notifications par liste.

```
define contactgroup{
   contactgroup_name   admins
   alias               admins
   members             admin,tech-qiminfo
}
```

* *Figure 32 Ajout d'un groupe de contact*

e) Ajouter une commande

Les commandes se trouvent dans /etc/shinken/commandes et /etc/shinken/packs pour les commandes des packs installés. Chaque élément surveillé est contrôlé par une commande qui définit l'utilisation d'un plugin et les paramètres. Dans une commande on doit définir le nom de la commande et la commande à exécuter. Cette commande est la syntaxe de l'appel du plugin. Avec l'installation de Shinken et packs nous avons toutes les commandes disponibles. Mais on peut faire des commandes sur mesures. Dans une commande on peut directement passer les macros, variables ou arguments d'un hôte ou service.

```
## Check a TCP port
# This plugin tests TCP connections with the specified host (or unix socket).
# check_tcp -H host -p port [-w <warning time>] [-c <critical time>] [-s <send
# string>] [-e <expect string>] [-q <quit string>][-m <maximum bytes>] [-d
# <delay>] [-t <timeout seconds>] [-r <refuse state>] [-M <mismatch state>]
# [-v] [-4|-6] [-j] [-D <warn days cert expire>[,<crit days cert expire>]] [-S
# <use SSL>] [-E]
define command {
   command_name    check_http_nagios
   command_line    $NAGIOSPLUGINSDIR$/check_http -I $HOSTADDRESS$ $ARG1$
}
```

Figure 33 Commande check http

Exécutable avec son chemin macro argument

f) Ajouter Template

Shinken fournit par défaut des templates génériques pour la définition d'un host, service, timeperiode et contact. Dans ce templates certains paramètres comme les intervalles de check et notifications sont définis. L'utilisation de ces templates se fait par le champ use dans une définition d'un service, hôte, contact ou autre. L'installation des packs fournit également des templates. Un template peut inclure uniquement une seule définition comme un service, un hôte ou plusieurs en même temps. Les templates des packs comprennent souvent une définition d'hôte et des services liés. J'ai utilisé souvent les templates générique de base fournit par Shinken. On peut aussi générer des template pour faciliter les ajouts des éléments surveillés.

```
define host{
        name                            generic-host

        # Checking part
        check_command                   check_host_alive
        max_check_attempts              3
        check_interval                  3
        retry_interval                  2
        # Check every time
        active_checks_enabled           1
        check_period                    24x7

        # Notification part
        # One notification each day (1440 = 60min* 24h)
        # every time, and for all 'errors'
        # notify the admins contactgroups by default
        contact_groups                  admins,users
        notification_interval           1440
        notification_period             24x7
        notification_options            d,u,r,f
        notifications_enabled           1

        # Advanced option. Look at the wiki for more informations
        event_handler_enabled           0
        flap_detection_enabled          0
        process_perf_data               1

        # Maintenance period
        #maintenance_period             workhours

        # Dispatching
        #poller_tag                      DMZ
        #realm                           All

        # For the WebUI
        #icon_set                        server ; can be database, disk, network_service, server

        # This said that it's a template
        register                        0
}
```

* *Figure 34 Template generic-host*

g) Ajouter timeperiode

Les périodes sont utilisés pour déterminer quand les sondes et les notifications auront lieu. On peut ne pas sonder et notifier certaines services ou hôtes pour certaines périodes. On peut définir des heures du travail, des vacances ou des jours spéciaux. Je surveille un PC dans le parc informatique et ce PC est éteints en dehors des heures du travail. En l'occurrence il ne faut pas sonder ni notifier en dehors des heures du travail. Le template qui définit les heures du travail du lundi au vendredi pour régle ce problème.

```
define timeperiod{
        timeperiod_name                 24x7
        alias                           Always
        sunday                          00:00-24:00
        monday                          00:00-24:00
        tuesday                         00:00-24:00
        wednesday                       00:00-24:00
        thursday                        00:00-24:00
        friday                          00:00-24:00
        saturday                        00:00-24:00
}
```

** Figure 35 définition d'un timeperiode 7j/7 24h/24*

```
define timeperiod{
        timeperiod_name workhours
        alias           Normal Work Hours
        monday          09:00-17:00
        tuesday         09:00-17:00
        wednesday       09:00-17:00
        thursday        09:00-17:00
        friday          09:00-17:00
        }
```

** Figure 36 Définition d'un timeperiode pour les heures du travail*

h) Ajouter dépendance de host

La définition des dépendances est assez vitale pour un système de supervision car la corrélation des données dépend fortement de dépendances.
La définition de dépendances permet d'enlever des check et notifications inutiles qui compliquent les tâches des administrateurs. Ces définitions sont ajoutées suite à l'établissement de dépendances dans le chapitre dépendances 4.2 . Dans le cas du serveur chez OVH la vérification de celui-là dépend des autres éléments qui se situent avant le serveur comme l'équipement du réseau. Le système détermine lancement d'un check ou d'une notification avec la définition d'une dépendance. Pour chaque définition on détermine des critères pour les check et notifications. Les critères sont :

- **o** = échec de l'état UP
- **d** = échec de l'état DOWN
- **u** = échec de l'état UNREACHABLE
- **p** = échec de l'état pending (le check n'as pas encore eu lieu)
- **n** (none) : dépendance est toujours vérifié donc pas de critères

Avant chaque check ou notification il va d'abord regarder si les critères de notification et check sont bons par rapport à la définition. Si les critères ne sont pas appliqués le check et/ou la notification sont annulés. J'ai utilisé les critères d, u, p, cela veut dire que si l'un des hôtes est dans un état down, unreachable ou pending les hôtes dépendants ne vont pas être contrôlé. Si le routeur qgertr002 est inaccessible le Shinken ne contrôle pas le serveur (extsrv003).

```
define hostdependency{
      host_name                      Acces_Internet,qgeisp002,qgeswi002,qgertr002,qgeesx005,qgemnt001
      dependent_host_name            extsrv003
      execution_failure_criteria     d,u,p
      notification_failure_criteria  d,u,p
      dependency_period              24x7
 }
```

** Figure 37 Définition d'une dépendance*

i) Ajouter dépendances de service

La définition de dépendance de service est faite de la même manière que la dépendance d'hôte. On doit définir les critères pour les notifications et check.

- **o** = échec de l'état OK
- **w** = échec de l'état WARNING
- **u** = échec de l'état UNKNOWN
- **c** = échec de l'état CRITICAL
- **p** = échec de l'état pending (le check n'as pas encore eu lieu)
- **n** (none) : dépendance est toujours vérifié donc pas de critères

```
define servicedependency {
       host_name                        extsrv003
       service_description              ApacheHTTP -Etat
       dependent_host_name              extsrv003
       dependent_service_description    intranet-UAT-HTTP
       execution_failure_criteria       c
       notification_failure_criteria    c
       dependency_period                24x7
       }

define servicedependency {
       host_name                        extsrv003
       service_description              MySQL-Etat
       dependent_host_name              extsrv003
       dependent_service_description    intranet-UAT-HTTP
       execution_failure_criteria       c
       notification_failure_criteria    c
       dependency_period                24x7
       }
```

** Figure 38 Définition d'une dépendance de service*

j) Ajouter un Business rule (Règles métier)

Une règle métier est déclarée comme un service dans /etc/shinken/services avec une commande spéciale.

```
# definition de la regle metier pour application intranet, acces depuis local de geneve
 define service{
        use                     generic-service
        host_name               extsrv003
        service_description     Intranet UAT interne
        check_command           bp_rule!(qgeswi002) & (qgertr002) & (qgeisp002)& (extsrv003,Memoire & (extsrv003,Espace Disque) & (extsrv003,CPU Charge moyen)
  &(Acces_Internet) & (extsrv003,MySQL-Etat) & (extsrv003,intranet-UAT-HTTP) & (extsrv003,intranet-UAT-HTTPS)
        check_interval          1
        retry_interval          1
        business_impact         4
 }
```

L'importance pour le business   ET   service et son hôte

Figure 39 Définition d'une règles métier

5.1.3. *Supervision Sans agent via SSH*

Un check en SSH a besoin une connexion SSH via une clé sans mot de passe. Le service sshd est installé par défaut sur des machines linux mais pour certaines machines il faut activer ce service comme l'hyperviseur Esxi.
Pour la machine linux distante:
[shinken @qgemnt001] # service sshd start
Pour l'hyperviseur Esxi :Selon procédure d'activation ssh en annexe D.
La librairie python-paramiko doit être installée pour la connexion ssh.
[hbi@qgemnt001] # sudo yum install python-paramiko

Pour lancer des commandes de mesure il faut une connexion ssh donc une clé ssh
[shinken @qgemnt001] #ssh-keygen
Il faut envoyer la clé vers la machine distante suite à la création de l'utilisateur shinken .
[shinken @qgemnt001] #ssh-copy-id –i ~/.ssh/id_rsa shinken@ipserveur
Cet utilisateur shinken doit avoir le droit(644) de lecture sur clé sur la machine à surveiller. La clé se trouve ~/.ssh/authorized_keys, pour l'utilisateur concerné.
Tester la connexion en SSH
[shinken @qgemnt001] # ssh –i /.ssh/id_rsa shinken@IP_hôte
Suite à la réussite en connexion SSH, on peut soit utiliser le pack linux-SSH soit déclarer tous les services et paramètres manuellement. Pour avoir un contrôle total j'ai utilisé la solution manuelle mais j'ai profité des packs pour créer mes services. Les commandes de ces packs ont été utiles aussi.
Après le premier test j'ai eu un problème avec le check cpu en SSH qui donne le message d'erreur suivante :
« can not fetch stats from cpu »
Solution :

1- Vérification du paquet sysstat sur la machine à surveiller- OK
2- Le plugin est fait pour des systèmes en anglais alors que la machine est en français. J'ai modifié le plugin pour rendre compatible avec français.

[shinken @qgemnt001]# vi /var/lib/shinken/libexec/check_load_average_by_ssh.py
Ligne 81 : Remplacer (not line.startswith('Average')) par (not line.startswith('Moyenne'))
Il vaut mieux garder deux versions de ce fichier.

5.1.4. *Supervision via SNMP*

La surveillance par le protocole SNMP a besoin des outils SNMP installés sur le serveur de la supervision.
[hbi@qgemnt001] # sudo yum -y install net-snmp net-snmp-utils
Les outils SNMP nous aident à tester la communication avec un élément à surveiller en ligne de commande avant de l'ajouter dans supervision.

Si la machine à surveiller n'as pas le protocole SNMP installé, il faut l'installer.
Installation et configuration du protocole SNMP pour une machine Centos

[root@machine_distant] # yum instal snmp
[root@qgemnt001]# chkconfig snmpd on

Modifier config pour interroger le host
[root@machine_distant] # vim /etc/smp/smpd.conf
Ajouter les lignes suivantes:
agentAddress udp:0.0.0.0:161
rocommunity ***nom_de_community***
Ce nom de community est un plus pour la sécurité car sans avoir la connaissance du mot, on ne peut pas interroger la base MIB.
[root@machine_distant] # /etc/init.d/snmpd restart
On peut tester le service snmpd sur cette machine
[root@qgemnt001] # netstat -an |grep -i udp | grep 161

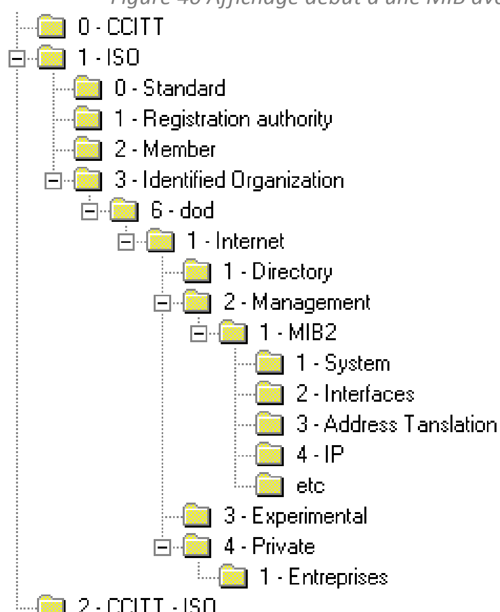
Dans le cas des équipements réseau on peut faire cette opération avec une connexion Telnet si une connexion est disponible. La méthode plus facile est d'aller faire cela directement sur l'interface web de gestion. Les équipements professionnel comme le routeur Cisco et Switch Netgear supporte la gestion SNMP via leurs interface web.
Test l'ouverture du port 161 depuis le serveur shinken
[shinken@machine_distant] # nmap Ip_machine_distant –p 161

Et scanner le port udp 161 -> OK

On peut tester directement les données obtenue via la commande :
[shinken@qgemnt001] # snmpwalk -v 2c -c public IP_machine_supervisée -> OK
OID   information

```
[root@qgemnt001 shinken]# snmpwalk -On -v2c -c [redacted] 192.168.10.54
.1.3.6.1.2.1.1.1.0 = STRING: GS724Tv3
.1.3.6.1.2.1.1.2.0 = OID: .1.3.6.1.4.1.4526.100.4.17
.1.3.6.1.2.1.1.3.0 = Timeticks: (132415804) 15 days, 7:49:18.04
.1.3.6.1.2.1.1.4.0 = STRING:
.1.3.6.1.2.1.1.5.0 = STRING: SW03
.1.3.6.1.2.1.1.6.0 = STRING: Armoire
```

** Figure 40 Affichage début d'une MIB avec snmpwalk*



Les plugins standards permettent de surveiller les équipements standards. Ces plugins interroge la base de MIB et récupère directement le facteur voulu avec l'identificateur de l'OID.
Mais dans le cas des équipements pas très répandus, nous sommes obligés d'aller trouver l'identificateur OID correspondant dans la base de MIB. Cette base est volumineuse et difficile à chercher une information précieuse. Mais il existe des outils comme Oidview pour visualiser la base d'un équipement facilement. Le plugin check_snmp permet de récuperer une information avec un argument OID.
Après avoir récupéré l'identificateur on peut récupérer l'information en passant comme paramètre au plugin check _snmp.

**   Figure 41 La hiérarchie de la MIB obtenue avec Oidview*

5.1.5. Supervision applicative

a) URL

Une requête http permet de tester la disponibilité d'un site internet. Le résultat de ce check est la réponse reçu du site. Un site peut être aussi texté avec une requête https si le site comporte un certificat SSL.

```
#Pour verifier url de l'application de SVN jira
define service{
        use                         generic-service
        host_name                   extsrv003
        service_description         jira-UAT-HTTP
        check_command               check_url !http://jira.[redacted].net
        check_interval              5
        retry_interval              1
}
```

* Figure 42 Service check URL jira

```
define command {
    command_name     check_url
    command_line     $NAGIOSPLUGINSDIR$/check_url_status_1.3 -U  $ARG1$
}
```

* Figure 43 Commande check URL

b) Check FTP

Un check pour le service FTP existe dans les plugins de Nagios. Il fait une requête ftp pour vérifier la disponibilité du serveur FTP.

```
define service{
  service_description    ftp
  use                    generic-service
  host_name              extsrv003
  check_command          check_ftp
}
```

* Figure 44 service FTP

```
define command {
       command_name      check_ftp
       command_line      $NAGIOSPLUGINSDIR$/check_ftp -H $HOSTADDRESS$
}
```

** Figure 45 Commande check FTP*

c) MySQL

Le plugin va simuler une connexion MySQL avec un compte crée sur le serveur distant pour Shinken.

```
define service{
  service_description     MySQL-Etat
  use                     generic-service
  register                1
  host_name               extsrv003
  check_command           check_mysql_connection_basic !178.xx.xx.xx !xxxx !XXXXX
}
```

** Figure 46 Service MySQL*

```
# Distant mysql check BASIC
define command {
       command_name      check_mysql_connection_basic
       command_line      $PLUGINSDIR$/check_mysql --hostname $ARG1$ --username $ARG2$ --password $ARG3$
}
```

* *Figure 47 Commande check MySQL*

d) DNS

Un plugin vérifie l'entrée d'une url dans la base de DNS.

```
define service{
       use                          generic-service
       host_name                    extsrv003
       service_description          jira-entree-DNS
       check_command                check_dns_entry
       _HOSTDNSHOSTNAME             jira.xxx.net
       _HOSTDNSEXPECTEDRESULT       178.xx.xx.xx
}
```

** Figure 48 service entrée DNS*

```
define command {
      command_name      check_dns_entry
      command_line      $PLUGINSDIR$/check_dns -H $_HOSTDNSHOSTNAME$ -a $_HOSTDNSEXPECTEDRESULT$ -t 120
}
```

** Figure 49 Check commande DNS*

e) Serveur Apache

Le plugin va chercher des informations via la page statuts du service httpd.

```
# check apaceh stat page et OK si page est accesible avec apaceh status OK
define service{
  service_description    ApacheHTTP -Etat
  use                    linux-ssh-service
  register               1
  host_name              extsrv003
  check_command          check_apache_stat !-p 443
  $HOSTADDRESS$          178.xx.xx.xx
}
```

** Figure 50 Service serveur Apache*

```
define command{
 command_name    check_apache_stat
 command_line    $NAGIOSPLUGINSDIR$/check_apachestatus.pl -H $HOSTADDRESS$ $ARG1$
}
```

** Figure 51 Check commande Apache*

5.2. INSTALLATION GRAPHITE

Installation de dépendances :
[root@qgemnt001] # yum install -y graphite-web* python-carbon* pycairo* python3-cairo* node npm mod_wsgi
[root@qgemnt001] # pip install 'django<1.6'
[root@qgemnt001] # pip install 'Twisted<12'
[root@qgemnt001] # pip install django-tagging
[root@qgemnt001] # pip install whisper
[root@qgemnt001] # pip install graphite-web
[root@qgemnt001] # pip install carbon

Creation de la base de données :
[root@qgemnt001] # /opt/graphite/webapp/graphite/ python manage.py syncdb
Configuration

[root@qgemnt001] # cp /opt/graphite/examples/example-graphite-vhost.conf /etc/httpd/conf.d/graphite.conf
[root@qgemnt001] # cp /opt/graphite/conf/storage-schemas.conf.example /opt/graphite/conf/storage-schemas.conf
[root@qgemnt001] # cp /opt/graphite/conf/storage-aggregation.conf.example /opt/graphite/conf/storage-aggregation.conf
[root@qgemnt001] # cp /opt/graphite/conf/graphite.wsgi.example /opt/graphite/conf/graphite.wsgi
[root@qgemnt001] # cp /opt/graphite/conf/graphTemplates.conf.example /opt/graphite/conf/graphTemplates.conf

Démarrage de Graphite :
[root@qgemnt001] # systemctl restart httpd
[root@qgemnt001] # /opt/graphite/bin/carbon-cache.py start
[root@qgemnt001] # /opt/graphite/bin/run-graphite-devel-server.py /opt/graphite/

Graphite ne démarre pas correctement à cause des problèmes de droits :
IOError: [Errno 13] Permission denied: '/opt/graphite/storage/*'
Pour régler tous les problèmes liés aux droits.
[root@qgemnt001] chown -R apache:apache /opt/graphite/storage/
[root@qgemnt001] chmod -R a+w /opt/graphite/storage
[root@qgemnt001] chmod 644 /opt/graphite/storage/*
[root@qgemnt001] sudo chmod a+w /opt/graphite/storage/log/webapp

Procédure d'installation Graphite :
http://www.rampmeupscotty.com/blog/2012/08/07/installing-graphite-on-centos-6-dot-2/
https://mespotesgeek.fr/des-graphiques-sur-shinken-via-graphite/

5.2.1. *Jonction à Shinken*

On installe le module Graphite depuis les dépôts Shinken
[shinken@qgemnt001] # shinken install ui-graphite
[shinken @qgemnt001] # shinken install graphite
On ajoute le module graphite-ui à la WebUI :
[shinken @qgemnt001] # vi /etc/shinken/modules/webui.cfg
et ajouter : **modules auth-cfg-password,SQLitedb,ui-graphite**

On ajoute le module graphite au broker :
[shinken @qgemnt001] # vi /etc/shinken/brokers/broker-master.cfg
et ajouter : modules webui,graphite

On configure le module graphite :
[shinken @qgemnt001] # vi /etc/shinken/modules/graphite.cfg

```
define module {
   module_name      Graphite-Perfdata
   module_type      graphite-perfdata
   host             localhost
   port             2003  ; Or 2004 if using use_pickle 1
}
```

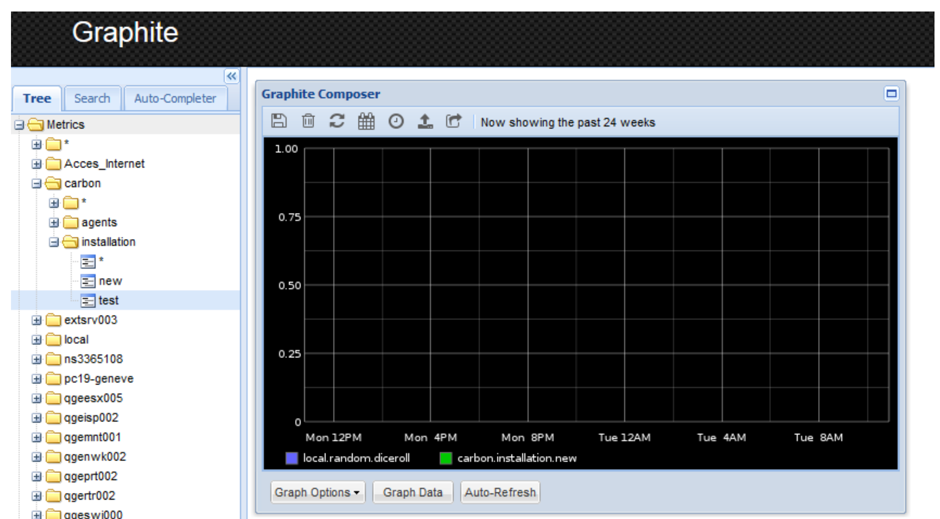
** Figure 52 Configuration du module Graphite*

On configure l'accès à la webui graphite sous /etc/shinken/modules/ui-graphite.cfg

```
define module {
    module_name     Graphite-UI
    module_type     graphite-webui
    uri             http://192.168.10.75:8080 ; Set your Graphite URI. Note : YOURSERVERNAME will be
                                         ; changed by your broker hostname
   templates_path  /var/lib/shinken/share/templates/graphite/
}
```

** Figure 53 Configuration Interface Web Graphite*

Si on met slash à la fin de l'URL les liens ne fonctionneraient pas.On rédemarre tout en tant que root.
[root@qgemnt001] # systemctl start httpd
[root@qgemnt001] # /opt/graphite/bin/carbon-cache.py start
Pour commencer on peut lancer le Graphite en mode développement pour voir ce qui se passe.
[root@qgemnt001] # /opt/graphite/bin/run-graphite-devel-server.py /opt/graphite/
On relance Shinken pour prendre en compte tout cela.
[shinken @qgemnt001] # /etc/init.d/shinken restart
Envoyer une donnée de test :
[root@qgemnt001] # echo "carbon.installation.test $RANDOM `date +%s`" | nc -w 1 localhost 2003
Les données envoyées doivent être disponible dans l'interface web de Graphite.



** Figure 54 Test de fonctionnement Graphite*

5.3. INSTALLATION THRUK ET NAGVIS

Suite à plusieurs essais d'installation par source, j'ai eu des problèmes liés aux dépendances de librairies Perl. Pour faire marcher certains plugins en Perl j'avais changé la version Perl du serveur, malgré la réinstallation de toutes les libraires manuellement le problème n'est pas résolu. L'installation par source est déconseillée pour l'installation de ces outils.
J'ai trouvé une distribution nommé OMD et qui contient les outils thruk, Nagvis, multisite et les outils de suppression. Ces outils doivent être couplés avec Shinken après l'installation.
Télécharger le paquetage depuis
http://files.omdistro.org/releases/centos_rhel/omd-1.20.rhel7.x86_64.rpm et l'outil s'install et il faut configurer.
Installer les package manquants.
[root@qgemnt001] # omd setup
Créer un site.
[root@qgemnt001] # omd create monitoring
Configurer le site et mettre les fichiers par défaut.
[root@qgemnt001] # omd init monitoring

Il faut configurer les outils.
[monitoring@qgemnt001] # omd config monitoring
Après la configuration le thruk et Nagvis sont disponibles sous les liens :
http://localhost/monitoring/thruk
http://localhost/monitoring/nagvis
Les interfaces sont vides car il faut ajouter le socket du module Livestatus du Shinken.
Thruk : aller dans Config Tool et Backends/sites pour ajouter Shinken avec le socket de connexion localhost :5000.
Nous passons l'adresse localhost cat tous les outils installé sur le même serveur mais pour récupérer des données depuis un Shinken à distance, il suffit de mettre son adresse IP publique.

Thruk Backends Manager
Note: some options are not accessible here for security reasons.
Backends Configuration
Name: External Shinken
Type: livestatus
Connection: 127.0.0.01:50000 test
Hidden: No Yes
Section: group sites by sections
add new connection
Save Changes

Figure 55 Config Tool de Thruk

Thruk fournit la fonctionnalité de « surveillance en temps réel » comme Shinken mais l'interface est compliqué et il sera utilisé pour toutes fonctionnalités que Shinken ne fournissent pas.

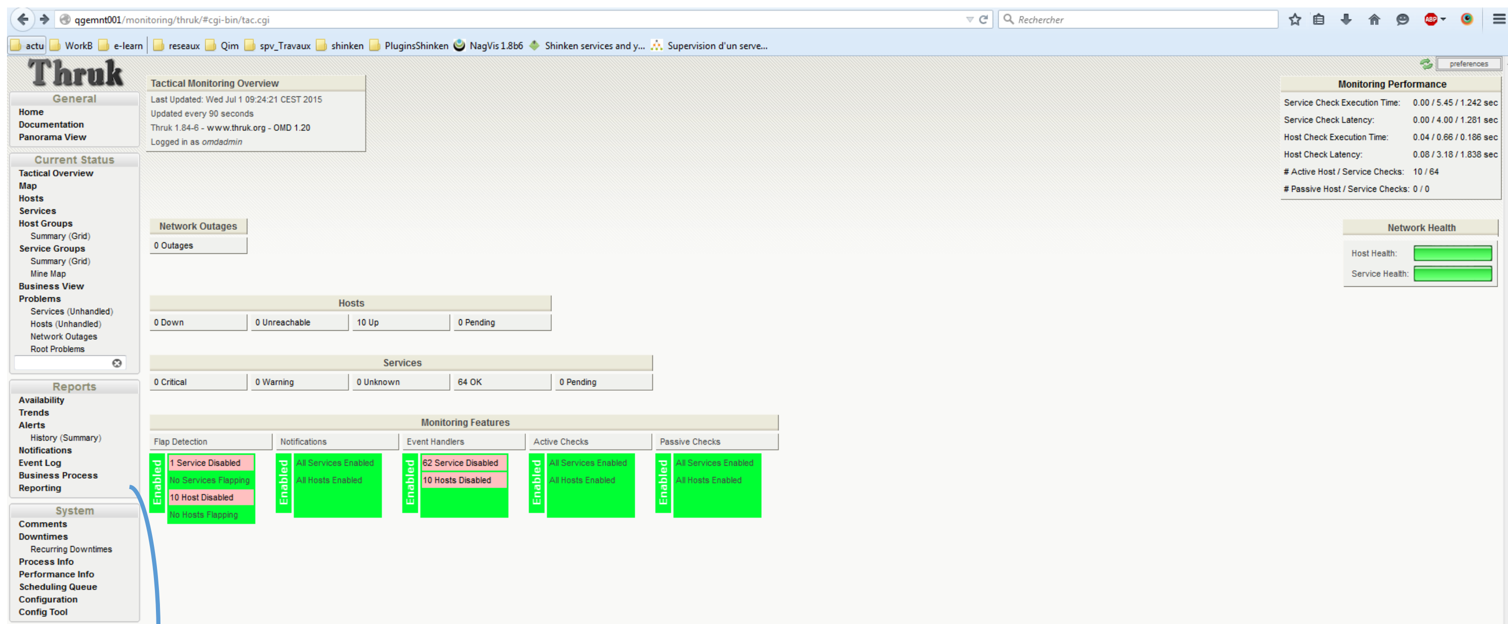
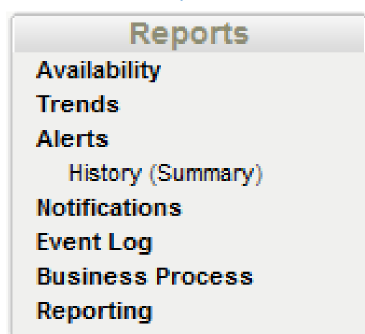


Figure 56 interface web de Thruk



Donc dans notre cas le Thruk sera utilisé plutôt pour faire le reporting pour :

- Availability :montre disponibilité des machines et services
- Trends : permet de grapher des données mesurables en fonction du temps
- Alerts: permet d'avoir un historique des alertes et des notifications
- Reporting : génère des rapports de SLA automatiques et manuels.

Historiques des pannes peuvent être visualisé dans ces différentes vues, on peut aussi générer un rapport avec la vue reporting.

Il peut être utile pour voir toute la configuration aussi. Car sous shinken pour voir les commandes ou les configurations d'un élément surveillé, il faut consulter les fichiers de Shinken. Alors que thruk nous affiche tout sur l'interface.

Hostgroup 'geneve' Host State Breakdowns:

Host	% Time Up	% Time Down	% Time Unreachable	% Time Undetermined
pc19-geneve	90.830% (90.830%)	9.170% (9.170%)	0.000% (0.000%)	0.000%
qgeesx005	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
qgeisp002	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
qgemnt001	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
qgenwk002	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
qgeprt002	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
qgertr002	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
qgeswi002	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
Average	98.854% (98.854%)	1.146% (1.146%)	0.000% (0.000%)	0.000%

Hostgroup 'ovh' Host State Breakdowns:

Host	% Time Up	% Time Down	% Time Unreachable	% Time Undetermined
extsrv003	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%
Average	100.000% (100.000%)	0.000% (0.000%)	0.000% (0.000%)	0.000%

** Figure 57 Rapport de disponibilité des machines sur une semaine*

	Edit Report
Name*	Rapport qgeswi002
Description	Rapport Mensuel pour siwtch Cisco
Public	(•) yes ( ) no
	E-Mail Settings:
To	
Cc	
Schedule	add more

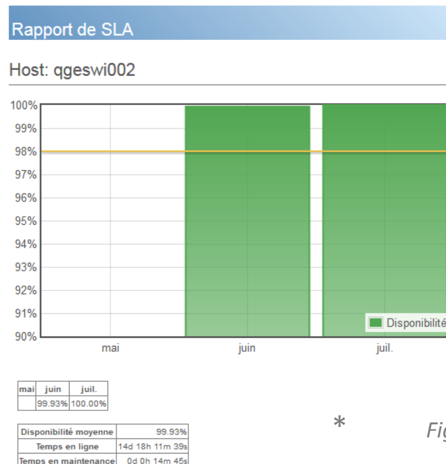
	Report Type
Type	SLA Host
	Report Options:
Language*	french
Host*	qgeswi002
Timeperiod	Last 31 Days
Breakdown by	Months
Report Timeperiod	None
SLA %*	98
Graph SLA %*	90
Details SLA %* hide details if sla is above threshold	-1
Decimal Points*	2
Assume Initial States	Yes
Initial Assumed State	Unspecified
Include Soft States	No
Unavailable States*	

State	Normal	During Downtime
Up	☐	
Down	☑	☐
Unreachable	☑	☐

** Figure 59 Génération d'un rapport*

Génération d'un rapport se fait via le menu reporting et create new report. Thruk nous permet de générer des rapports assez facilement et sur mesures. On peut programmer la génération d'un rapport avec une période et envoie par mail. Un rapport peut être fait pour

- Un service ou groupe de service
- Une machine ou un groupe de machines
- Toutes les pannes survenues



Un Rapport contient une graphique de disponibilité et les dernières interruptions.

** Figure 58 Rapport pour un switch*

Rapport de SLA

Dernières interruptions

16 jun 2015 17:31:06 - 16 jun 2015 17:33:44 (0d 0h 2m 38s)
CRITICAL - Host Unreachable (192.168.10.54)

16 jun 2015 16:54:33 - 16 jun 2015 16:58:45 (0d 0h 4m 12s)
CRITICAL - Host Unreachable (192.168.10.54)

16 jun 2015 16:26:26 - 16 jun 2015 16:32:23 (0d 0h 5m 57s)
CRITICAL - Host Unreachable (192.168.10.54)

16 jun 2015 15:26:10 - 16 jun 2015 15:28:08 (0d 0h 1m 58s)
CRITICAL - Host Unreachable (192.168.10.54)

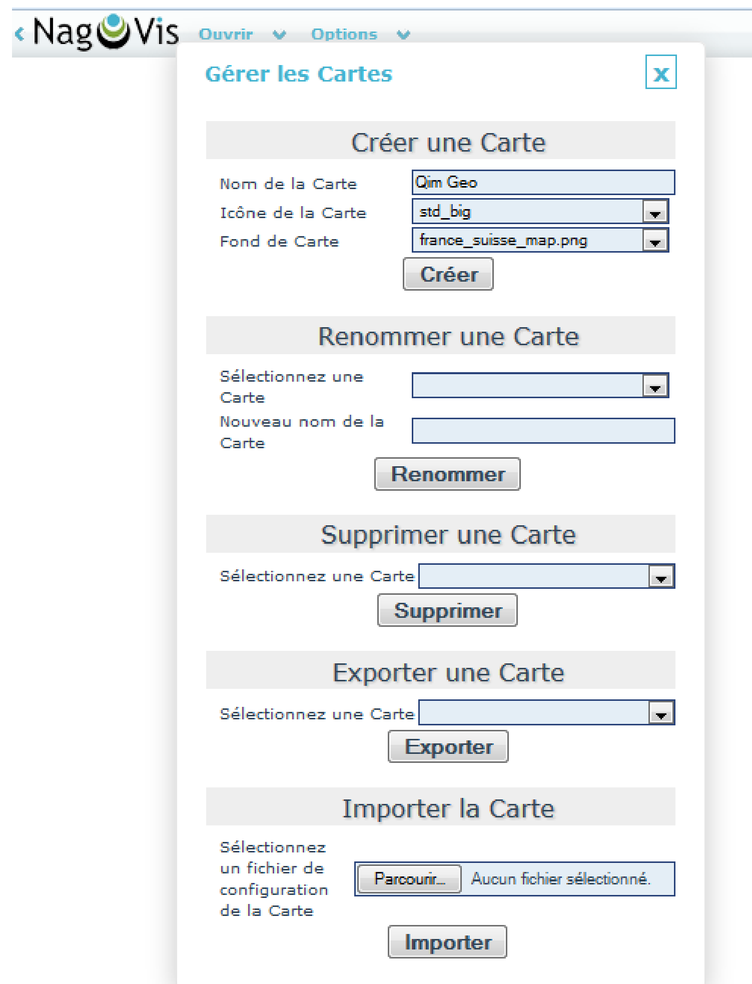
Nagvis gère la connexion avec Shinken de la même manière que le Thruk. On doit ajouter la connexion du Shinken via le menu option et gère les Backends.

** Figure 60 Ajouter Shinken dans Nagvis*

Pour créer des cartes nous devons envoyer des images sur Nagvis. J'ai utilisé les schémas et des photos de la salle de serveur pour créer des cartes.



* *Figure 61 ajout d'une image dans Nagvis*



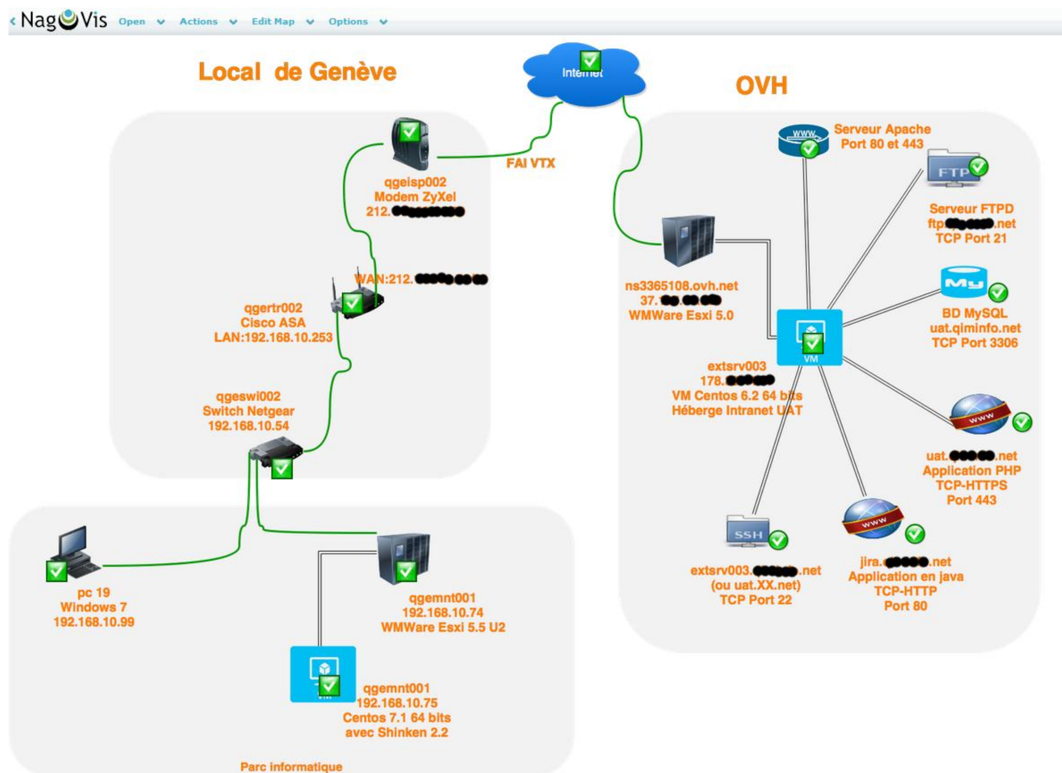
* *Figure 62 Créer une carte dans Nagvis*

Après la création des cartes, on peut ajouter les éléments surveillés dans ces cartes.



* *Figure 63 Ajouter un élément surveillé*

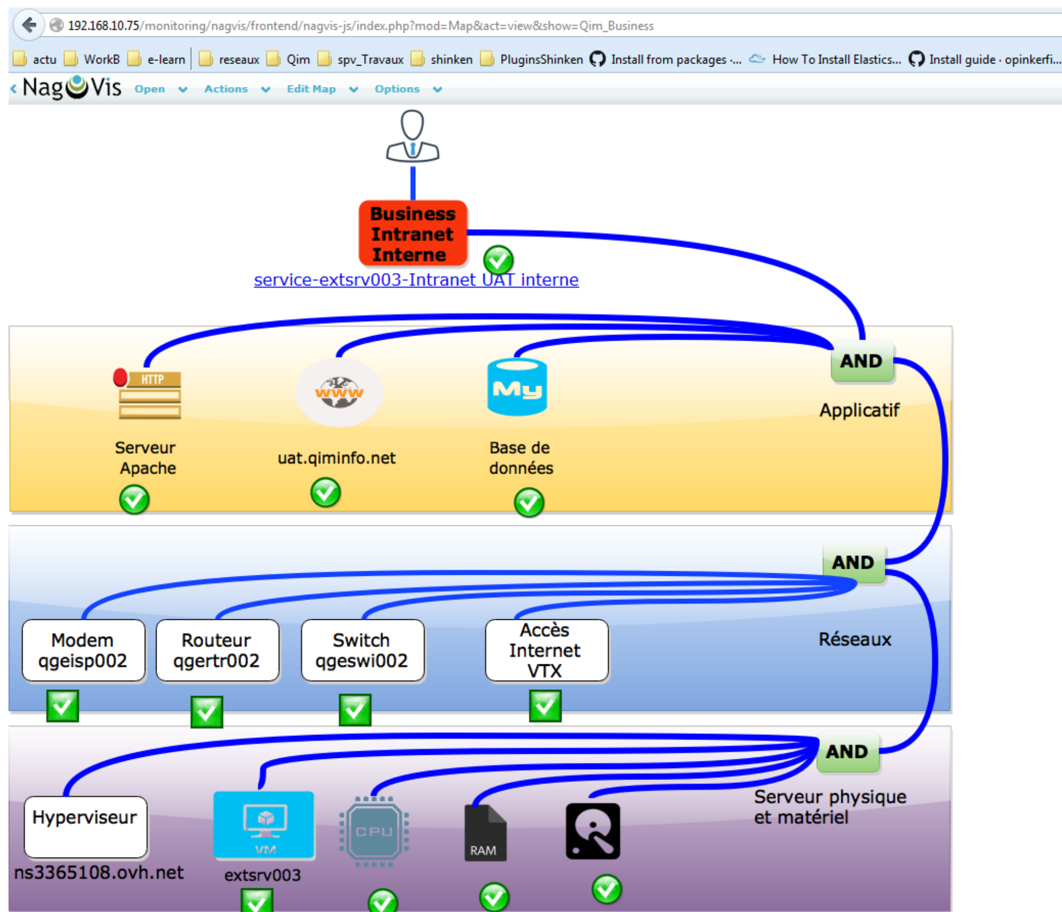
Nagvis permet d'ajouter une machine, un service, une règles métier et un élément peut être symbolisé avec un icône ou une ligne.



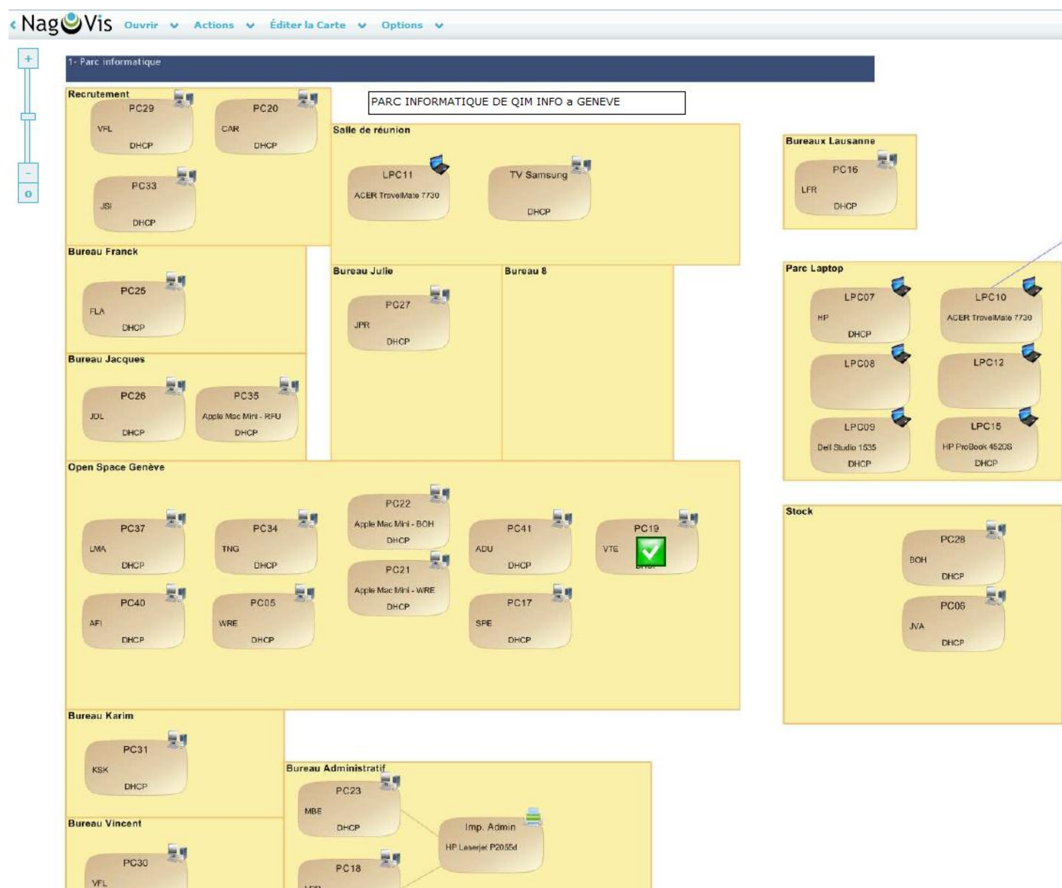
** Figure Carté Nagvis basée sur Schéma du scénario de l'utilisation de l'Intranet Chapitre 4*



** Figure 64 Carte de la salle de serveur*

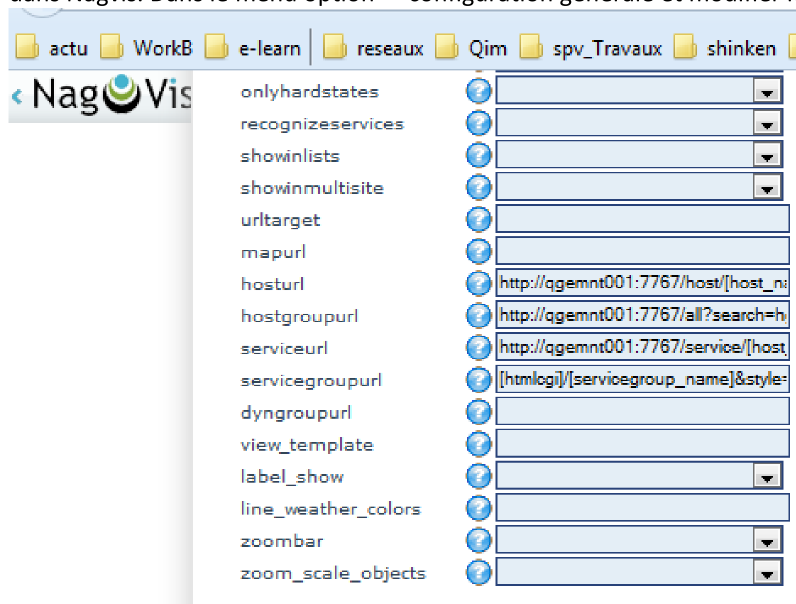


* *Figure Carté Nagvis basée sur règles métier du scénario de l'utilisation de l'Intranet Chapitre 4*



** Figure 65 Carte du parc informatique de Genève*

On doit configurer les urls dans Nagvis pour qu'on puisse aller sur l'interface Shinken pour un élément cliqué dans Nagvis. Dans le menu option -> configuration générale et modifier les templates des urls pour Shinken.



** Figure 66 direction url de Nagvis vers Shinken*

5.4. PROBLÈMES FRÉQUENTS.

Il est possible d'avoir des erreurs en cas de mauvaise configuration, manque de librairies ou problèmes avec les plugins. Si l'erreur viens du shinken il faut savoir que toutes les infos par rapport à ces erreurs sont enregistrées dans les fichiers log sous le répertoire */var/log/shinken/.* Il faut regarder dans broker.log, scheduler.log et arbiter.log.

5.4.1. *[Errno 2] No such file or directory*

Quand vous voyez ceci, c'est que la commande n'est pas présente dans le dossier **/var/lib/shinken/libexec/.** Je vous conseille de vérifier votre fichier de service (/etc/shinken/services/<fichier>.cfg) en vérifiant la ligne check_command.

Ensuite vérifier votre fichier de commande (/etc/shinken/commands/<fichier>.cfg) en vérifiant la ligne command_line.Et pour terminer, voir si la commande est bien présente dans le dossier /var/lib/shinken/libexec

5.4.2. *[Errno 13] Permission denied*

Cette erreur vous dis que Shinken n'a pas les permissions adéquates pour lancer les commandes dans le dossier */var/lib/shinken/libexec/*.
Voici comment résoudre ceci de manière sécurisée (STOP AU CHMOD 777)
Tout d'abord, on va redéfinir le propriétaire de la commande en erreur (pour savoir lequel :
ls -ail /var/lib/shinken/libexec) :
[root@qgemnt001]# chown shinken:shinken /var/lib/shinken/libexec/<fichier>

Puis nous allons redéfinir les droits d'accès de ce fichier en 755. Ce qui équivaut à :
Propriétaire : Lire, Écrire, Exécuter
Groupe : Lire, Exécuter
Publics : Lire, Exécuter

Revenons à nos moutons, la commande pour définir les droits est :
[root@qgemnt001]# chmod 755 /var/lib/shinken/libexec/<fichier>
Une fois fini, sélectionner le service sur l'interface web et faite "Recheck". Patientez une petite dizaine de secondes et rafraîchissez la page.

5.4.3. *Interface n'est pas accessible depuis le réseau local*

L'interface web est accessible via http://IP_du_serveur:7767/ mais le port était bloqué pour l'accès externe depuis le réseau. J'ai d'abord ajouté une règle dans le firewall de l'hyperviseur (qgeesx005) malgré cela l'accès était toujours bloqué car c'est le firewall du serveur de la supervision qui bloquait l'accès. J'ai ajouté une règle pour le port 7767 avec le protocole TCP.
[root@qgemnt001]# firewall-cmd --new-zone=special
[root@qgemnt001]# firewall-cmd --zone=special --add-source=192.168.10.0/24
[root@qgemnt001]# firewall-cmd --zone=special --add-port=7767/tcp

5.4.4. Adresse IP dynamique

Les équipements réseaux et les serveurs ont des IP fixes mais les postes de travail surveillés ont des IP dynamique donc en cas de supervision avec une adresse IP le changement d'IP est un grand problème. Pour contourner celui-là j'ajoute les postes avec leur nom de l'annuaire de l'Active Directory.

5.4.5. Pip broken

[root@qgemnt001]# yum remote python-pip

Si cela ne marche pas :

[root@qgemnt001]# rm /usr/lib/python2.7/site-packages/pip*

Réinstaller le pip :

[root@qgemnt001]# yum reinstall python pip

5.4.6. Problème libraire Perl

Comme il n'y pas mal de plugins en Perl, on a souvent affaire au problème de "C'ant locate *.pm ». Souvent l'erreur vient le manque d'une librairie ou incompatibilité avec version Perl. Dans le cas de la mande d'une librairie on peut l'installer via CPAN.

[hbi@qgemnt001]# sudo cpan nom_libraire

Les plugins en Perl ou python incluent le chemin du Perl ou Python au début du script, des fois les chemins sont mal renseignés et le plugin ne s'exécute pas. Il faut corriger les chemins. Vue la diversité des outils ou des plugins installés, il peut y avoir un problème de comptabilité des libraires. Comme cela a été le cas avec Thruk. On peut installer les différentes versions du Perl dans des répertoires séparés et renseignés les chemins de celles-ci dans les programmes qui les utilisent. Suite à nombre problèmes que j'ai eu, j'ai décidé d'installer le Thruk avec un package pour résoudre ces problèmes. Car une installation par package automatise l'installation des dépendances.

6. Bilan du projet

6.1. CONCLUSION TECHNIQUE

Les objectifs définis dans le cahier de charges (chapitre 2) ont été atteints. Shinken permet de superviser tous les éléments définis dans le scénario sauf l'hyperviseur nsXX.ovh.net car à cause d'un problème technique, il n'est pas accessible.
Le choix du Centos comme système d'exploitation n'a pas facilité la mise en œuvre. Ce choix est un choix historique pour l'entreprise mais la distribution Debian gère les dépendances d'une manière plus automatisée que Centos. Et j'ai trouvé plus de package pour Debian concernant les outils de la supervision. La difficulté technique est liée aux dépendances des librairies et aux installations des outils.
Le choix de l'outil a été correct car le moteur de Shinken est assez performant et modulaire, j'ai pu constater tous ces spécifications définis dans l'étude l'outil (Chapitre 3). Le surplus de Shinken par rapport à Nagios est remarquable surtout pour la supervision du business. Et les vues de l'interface web de Shinken sont assez compréhensibles pour les responsables de l'entreprise qui ne sont pas forcément des

informaticiens. L'architecture distribuable de Shinken permet de superviser des grandes infrastructures. L'installation et le paramétrage du shinken n'est pas très compliqué contrairement à ce que je croyais. Grâce à dépendances déclarées les notifications sont intelligentes, on ne reçoit des notifications que pour les sources des problèmes. Le superviseur est proactif grâce au seuil de « warning » car on reçoit une notification avant l'état critique avec ce seuil.
Le reporting par Thruk fournit des services assez intéressants car la disponibilité du système informatique sera visualisée facilement. Le système de reporting est assez compréhensible avec des graphiques. Mais le côté bénéfique du reporting sera visible une fois que le système est étendu sur l'infrastructure.
Le couplage de plusieurs outils nous donne une bonne solution complète car à la fois on peut visualiser le système supervisé sur une interface simple, moderne et claire et on peut voir des graphiques de mesures temporelles dans celle-là. Le thruk nous génère des rapports simple et efficace de la manière automatique programmé ou manuels quand on en a besoin. Les cartes de Nagvis rendent l'affichage en temps réel très pratique et facile.
La manière de ce couplage nous permet de changer ou modifier chaque outil indépendamment. Dans le cas d'une supervision distribuée le système peut s'adapter facilement.
Un autre aspect important est l'overhead crée comme charge réseau. Les latences et durée des sondes affichés dans l'interfaces web restent entre 0,1 et 3 secondes.
Auprès des captures faits avec Wireshark (Annexe I) le volume d'une sonde reste entre dans l'ordre de quelques octets. Taille d'un sonde pour FTP : 1,1 ko, SNMP 3,6 ko, SSH 6,1 ko, Windows 12 ko. Au total nous avons 64 services et 10 hôtes supervisés donc le volume total des sondes est moins d'un mégaoctet. Le 99% de ces 74 sondes lancés chaque 3 minutes. En moyenne cela ne représente pas plus que quelques centaines d'octets d'overhead par minute. Cet overhead reste assez raisonnable avec moins de 1% du Traffic réseau de l'entreprise.
La supervision devient un élément vital pour les entreprises qui disposent un système informatique fournissant des services.

6.2. Conclusion Personnelle

Ce projet formateur m'a permis d'approfondir mes connaissances techniques de l'administration d'une infrastructure. J'ai fait la connaissance avec le monde de la supervision et les outils de la supervision. Au-delà de l'enjeu technique, un des véritables acquis était la gestion du projet. Nous avons fait des projets tout au longs de nos études mais la question « comment bien mener un projet ? » a été concrétisé avec cette expérience. L'environnement de la réalisation m'a permis de connaitre la vie professionnelle et la discipline de travail que j'ai suivi au sein de l'entreprise m'aidera beaucoup pour intégrer le monde du travail. En suivant les procédures internes à l'entreprise j'ai appris les étapes de réalisation d'un projet dans un cas réel.

Je peux dire, sans retenue, que je suis pleinement satisfait de cette aventure.

6.3. Difficultés rencontrés

Avant de commencer ce projet je n'avais aucune idée sur le monitoring donc l'initialisation du projet a été un peu difficile car ce travail m'oblige à être beaucoup plus rigoureux par rapport à un projet de développement. Au début j'ai eu des difficultés au niveau de l'infrastructure de l'entreprise car le schéma existant montre le fonctionnement mais il n'est pas détaillé techniquement.
La difficulté majeure ne reposait pas sur Shinken car j'ai pu régler facilement les problèmes que j'ai eus avec Shinken mais elle reposait sur les checks et les outils couplés avec Shinken.
Le plus grand problème technique était au niveau des dépendances des libraires. J'ai testé et utilisé beaucoup de plugins avec des langages différents et des fois il y avait des incompatibilités entre les versions des librairies. Certains plugins ne marchaient pas avec les nouvelles versions des librairies comme le check_http de Shinken qui ne marchait pas avec openssl 1.0.1. A la place de celui-là j'ai utilisé le plugin de check_http de Nagios.
L'installation de la Graphite a été difficile aussi. Au début j'ai utilisé le script d'installation fourni par l'équipe du shop.ch mais cela n'as pas marché et j'ai utilisé les procédures d'installations que j'ai trouvé dans des blogs

(Installation Graphite chapitre 5.2). Il a fallu regarder le plus près à chaque composant comme carbon, webapp ou module Graphite de Shinken et réinstaller et tester chacun à part.
L'installation des outils de reporting a été problématique aussi. Surtout le thruk ne démarrait pas à cause des problèmes de dépendances. Il ne fallait pas changer les libraires Perl existant pour le Centos 7.1 car malgré l'installation de toutes les libraires via CPAN ou yum le problème n'était toujours pas résolu. La solution ultime a été d'installer les outils via un package spécial qui incluait les libraires nécessaires.
Après l'installation de ces outils la connexion entre shinken et ces outils ont provoqué des soucis mineurs. Le thruk n'arrivait pas à récupérer les données du Shinken avec un socket Unix et le module Livestatus du Shinken était cassé. J'ai réinstallé le module Livestatus et la connexion a marché avec un socket tcp.

6.4. RESTAURATION DU SERVEUR DE SUPERVISION

La VM est exporté en tant que template OVF (open virtulization format) et il prêt à être déployé sur une machine Esxi via le vSphere client. Les fichiers de configuration de Shinken sont sauvegardés dans le SVN de l'entreprise.

6.5. AXES D'AMÉLIORATIONS

Etendre le système sur l'infrastructure :
La salle des serveurs dans le local de Genève doit être surveillée en premier lieu car il y a de nombreux serveurs et d'applications hébergés. Comme la plupart des serveurs sont virtualisés une surveillance profonde de l'infrastructure virtuelle peut être bénéfique.
L'imprimante peut être surveillée pour détecter les niveaux des cartouches qui permettent de les remplacer avant que quelqu'un se rendre compte en essayant de faire une impression.
La deuxième sortie (switch + routeur+ modem + fournisseur Swisscom) pour l'accès internet doit être surveillé car certains applications et VPN en utilisent constamment.
Une autre partie importante qui doit être surveillé assez rapidement est les serveurs de productions chez OVH. Surtout une supervision de la couche applicative est nécessaire pour ces serveurs.
Supervision de la Couche applicative :
On peut surveiller la couche applicative plus profondément. La supervision actuelle de la couche applicative est plutôt la vérification de disponibilité des services mais on peut aller plus loin et surveiller la qualité de ces services, les états des processus, les threads, le temps de réponse, le nombre de connexion, les logs etc.
Le déploiement d'un Poller au bureau de Lausanne et chez OVH peut être prévu s'il y a beaucoup d'éléments à surveiller.
On peut même prévoir une installation Shinken ailleurs comme chez un client et regrouper celui-ci avec le superviseur existant via l'interface Thruk.
Supervision de la sécurité :

- Il existe des plugins qui permettent de détecter des ports ouverts (backdoor) qui peuvent être dangereux pour la sécurité.
- On peut ajouter la surveillance de ces backdoor sur les machines importantes.
- On peut aussi surveiller la sécurité des applications web contre les attaques. Un exemple sera la surveillance d'un hôte contre des attaques DDos.
- On peut contrôler les certificats SSL s'il est valide.
- On peut vérifier l'intégrité des documents importants avec des check MD5.
- Surveiller le VPN : le nombre d'utilisateur connecté, le tunnel de la connexion etc.
- On peut surveiller es attaques connues sur un pare-feu.

Notification : On peut ajouter le module Android ou utiliser des scripts faits pour Nagios pour envoyer des notifications vis SMS. Les périodes de notification et surveillance (timeperiod) peuvent être adapté encore mieux pour les postes de travail comme les périodes de vacances.

Gestion des incidents(Handler) : La supervision actuelle ne gère pas les incidents. Une gestion pour les incidents importants peut être très bénéfique pour l'entreprise. Certains incidents peuvent être résolus facilement comme redémarrage d'un service d'une VM.
Shinken a la possibilité d'exécuter une commande en cas de panne. Pour les machines ou services très critiques on peut créer des commandes à partir des plugins existants ou des scripts développés sur mesures. L'une de possibilité sera redémarrer une machine ou un service à distance.
Ou un autre exemple : Si le fournisseur d'accès internet tombe active deuxième FAI. Ou encore réveiller un serveur ou poste éteigne à distance via Wake on Lan.
Accès à l'interface web : L'accès est restreint au réseau local et il faut une connexion VPN pour pouvoir surveiller en temps réel mais on peut installer un site chez OVH pour avoir l'accès partout. Ce site va communiquer avec shinken pour avoir les données.
Ajout des éléments via interface Web : L'entreprise n'as pas une infrastructure très volumineuse et les ajouts des éléments peuvent se faire directement par fichier mais dans le cas où on doit ajouter beaucoup d'éléments, on doit prévoir une interface pour ajouter plus facilement. Il faut configurer Thruk pour pouvoir gérer les ajouts directement sur l'interface de thruk.
Pour l'instant l'outil gratuit convient très bien aux besoins de l'entreprise mais il faut savoir que Shinken possède aussi une version payante qui fournit quelques valeurs ajoutées comme :

- Système de reporting
- La gestion et configuration automatisé pour un système de supervision distribué
- Supervision de business avancé
- Détection des aberrations

Comme l'entreprise fournit des services de l'administration de l'infrastructure à ces clients, cette version peut être bénéfique pour des grands projets.

On peut ajouter des icônes dans Shinken afin de permet rendre l'affichage des équipements plus compréhensible et visible. Les icônes se trouvent dans /var/lib/shinken/modules/webui/htdocs/images/sets.

6.6. Cahier de tests

Les tests sont prévus pour chaque élément supervisé définis dans le scénario (chapitre 4.2) . Chaque test est validé avec trois critères. Le premier critère est l'affichage d'un problème simulé sur l'interface de Shinken. Deuxième critère est la réception d'un mail pour éléments surveillé. Et le dernier critère est la gestion de dépendances et corrélation de données. Les deux premiers critères ne posaient pas de souci mais le troisième critère a été validé après avoir reconfiguré les dépendances.
Quelques captures sont données en annexe H pour servir l'exemple.

Test case	Actions	Résultat attendu	Affichage sur WEBUI	Alerte Par Mail	Dépendances appliqués
)VH - Panne chez OVH sur serveur UAT(extsrv003)			**0 KO 0 non testés**	**0 KO 0 non**	**0 KO 0 non testés**
Serveur FTPD	Arrêter le serveur ftpd	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément mais aussi pour 4 règles métiers .	ok	ok	ok
Serveur Apache	Arrêter serveur apache	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément aussi pour 4 règles métiers . L'intranet UAT tombe en panne aussi mais on devrait pas checker ni envoyer alerte pour intranet avec	ok	ok	ok
Serveur MySQL pour BD	Arrêter serveur MySQL	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément et pour 4 règles métiers aussi. L'intranet UAT tombe en panne aussi mais on devrait pas checker ni envoyer alerte pour intranet avec	ok	ok	ok
Intranet UAT Interne	Arrêter application	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément mais aussi pour 1 règle métiers .	ok	ok	ok
Intranet UAT externe	Faire échouer le check en changeant le port de l'application de check. Le check se fait via un serveur	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément mais aussi pour 1 règle métiers .	ok	ok	ok
Jira UAT externe	Faire échouer le check en changeant le port de l'application de check. Le check se fait via un serveur	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément mais aussi pour 1 règle métiers .	ok	ok	ok
Jira UAT interne	Arrêter application	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément mais aussi pour 1 règle métiers .	ok	ok	ok
Connexion SSH	Arrêter SSH	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.	ok	ok	ok
CPU	Augmenter la consommation sur serveur	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément aussi pour 4 règles métiers .	ok	ok	ok
Disque	Créer des gros fichiers pour remplir le disque.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément aussi pour 4 règles métiers .	ok	ok	ok
RAM	Augmenter la consommation sur serveur en lançant des processus.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément mais aussi pour 4 règles métiers .	ok	ok	ok
/TX - Coupure d'accès internet- FAI VTX			**0 KO 0 non testés**	**0 KO 0 non**	**0 KO 0 non testés**
Enlever câble vdsl sur la prise murale.		Affichage de la panne sur WebUI et envoie d'alerte pour l'élément après cela devient OK. Pas d'alerte pour les éléments qui dépendent à l'accès internet. Pas d'alerte pour	ok	ok	ok

ISP - Panne sur modem qgeisp002			**0 KO 0 non testés**		**0 KO 0 non**		**0 KO 0 non testés**	
Eteindre le modem. Ou enlever le câble réseau.				ok		ok		ok
SWI -Panne sur switch net gear (qgeswi002)			**0 KO 0 non testés**		**0 KO 0 non**		**0 KO 0 non testés**	
Port 4	Enlever le câble.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.		ok		ok		ok
Port 5	Enlever le câble.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément après tout devient OK. La machine qgeesx0005 (héberge monitoring) est lié à ce port. Pas d'alerte pour les éléments qui dépendent à ce		ok		ok		ok
Port 6	Enlever le câble.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.		ok		ok		ok
Port 8	Enlever le câble.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.		ok		ok		ok
Port 9	Enlever le câble.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément. Ce port relie le switch au routeur. Pas d'alerte pour les éléments qui dépendent à ce port.		ok		ok		ok
Port 11	Enlever le câble.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.		ok		ok		ok
Port 12	Enlever le câble.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.		ok		ok		ok
Port 14	Enlever le câble.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.		ok		ok		ok
Port 17	Enlever le câble.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.		ok		ok		ok
Port 21	Enlever le câble.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.		ok		ok		ok
Port 23	Enlever le câble.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.		ok		ok		ok
Switch DOWN	Eteindre ou enlever le câble de la machine qgeesx005.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément. Tous les élément derrièer (extsrv,qgeswii002,qgeisp002 en panne mais pas d'alerte grâce aux		ok		ok		ok
RTR- Panne sur routeur Cisco ASA (qgertr002)			**0 KO 0 non testés**		**0 KO 0 non**		**0 KO 0 non testés**	
Interface eth0/0	Enlever le câble.	Affichage de la panne sur WebUI, alerte après avoir tout devient OK. Pas d'alerte pour les éléments plus loin que routeur déclarés comme		ok		ok		ok
Interface eth0/1	Enlever le câble.	Affichage de la panne sur WebUI, alerte après avoir tout devient OK. Pas d'alerte pour les éléments plus loin que routeur déclarés comme		ok		ok		ok
CPU w:85 c:95	Envoyer beaucoup de paquets.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.		ok		ok		ok

ESX - Panne sur serveur Esxi de monitoring (qgeesx005)			**0 KO 0 non testés**		**0 KO 0 non**		**0 KO 0 non testés**	
Connexion SSH.	Arrêter le service SSH.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément. Service ntp doit être en panne aussi car ntp est testé via une		ok		ok		ok
CPU-RAM w:85c:95	Charge la RAM avec des gros processus.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément. Possible retard et blocage.	**webui devient inaccessible en cas de 100%**	ok	**Retard pour envoie comme attendu.**	ok		ok
Disque w:98c:99 Disque	Remplir le disque avec des gros fichier.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément. Disque vm statique prends toute la place donc les seuils sont très		ok		ok		ok
Serveur NTP(Heure)	Arrêter NTP.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.		ok		ok		ok
Host DOWN.	Enlever câble réseau.	Pas d'accès WEbUI, envoie d'alerte après tout devient OK. Pas d'alerte pour les autres hôtes qui ne sont pas accessible à cause de cette panne.	**webui devient inaccessible en cas de 100%**	ok		ok		ok
/INTR - Panne sur VM de serveur de monitoring (qgemnt001)			**0 KO 0 non testés**		**0 KO 0 non**		**0 KO 0 non testés**	
CPU w:90 c:95	Charge la RAM avec des gros processus.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément. Possible retard et blocage.	**webui devient inaccessible en cas de 100%**	ok	**Retard 1-2 min pour envoie comme attendu.**	ok		ok
Disque w:85 c:95	Remplir le disque avec des gros fichiers.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément. Possible retard et blocage.		ok		ok		ok
RAM/swap w:90 c:90	Charge la RAM avec des gros processus. Script en C qui fait des alloc.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément. Possible retard et blocage.	**webui devient inaccessible en cas de 100%**	ok	**Retard pour envoie comme attendu.**	ok		ok
Host DOWN.	Faire échouer le ping en coupants bloquant la communications sur qgemnt001. Désactive icmp et enlèver câble réseau.	Pas d'accès sur webui, alerte après avoir tout devient OK.	**webui devient inaccessible**	ok	**Retard pour envoie comme attendu.**	ok		ok
PC 19- Panne sur le poste 19 dans le local de Genève (Pc19)			**0 KO 0 non testés**		**0 KO 0 non**		**0 KO 0 non testés**	
CPU w:90 c:95	Charge la RAM avec des gros processus.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.		ok		ok		ok
Disque w:90 c:95	Remplir le disque avec des gros fichiers.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.		ok		ok		ok
RAM w:90 c:95	Charge la RAM avec des gros processus.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.		ok		ok		ok
Host DOWN.	Enlever câble réseau.	Affichage de la panne sur WebUI et envoie d'alerte pour l'élément.		ok		ok		ok

* *Tableau 3 Cahier de tests*

Annexe A. Gestion de Projet

a. Déroulement et gestion du projet

La macro planning ci-dessous détermine les grandes dates du projet. Le cahier de charge et macro planning sont validés par le directeur technique de Qim info, Jacques DAUDEL.
Le projet est réalisé en 3 lots.
Le premier lot (Lot 0) est la partie étude pour étudier l'infrastructure de l'entreprise et la supervision.
Lot 1 est la grande partie de mise en œuvre du serveur de supervision. Ce lot valide aussi le bon choix de l'outil.
Après la première réussite de POC en lot 1, lot 2 permet de valider le système de supervision et ajouter d'autre outil comme outil de reporting. Comme la solution bêta est réalisée il faut bien s'assure que tous éléments surveillés sont correctement surveillés. Ce lot comprend principalement les tests avec divers scénarios et installation d'un produit pour avoir un historique de panne. Les tests d'acceptations sont essentiels pour ce lot.

i. Macro Planning

Je détermine les grandes étapes du projet avec une macro planning et les tâches sont définies dans le planning avec des estimations de durée.
Dates importantes : Début 20 avril
Remise du rapport à hepia 8 juillet
Présentation le 2 septembre
Etude : Etudier les outils, technologies, l'infrastructure etc.
Rédaction : Ecrire les documents.
Mise en œuvre : Installation, configuration, paramétrage et test.
Mises à jour : Appliquer le changement et correction après la démo.
Livrables sont les documents et le serveur de supervision à livrer à l'entreprise.

ETAPE	Semaine	Livrables	Activités
ÉTUDE RÉDACTION	1	LOT 0 Cahier de charges Comparaison des outils Initiation avec planning	Étude du système, solutions existantes Prendre en main la solution Établir planning Documentations
ÉTUDE MISE EN OEUVRE REDACTION	2	LOT1 Versions améliorés de ce document et un serveur avec l'outil installé si le choix est fait	Réunion avec consultant pour revoir planification Étude de la solution Installation du serveur de surveillance Configuration du serveur
ÉTUDE MISE EN ŒUVRE REDACTION	3	Un manuel d'installation de la solution	Installation et configuration su serveur supervision.
MISE EN ŒUVRE REDACTION	4	Un manuel de paramétrage de la solution	Activer les plugins et ajouter les éléments à surveiller.
MISE EN ŒUVRE	5		Ajout des éléments à surveiller.
MISE EN ŒUVRE	6	V 0.9 (Bêta)	Ajout des éléments à surveiller.

REDACTION			
ÉTUDE MISE EN ŒUVRE REDACTION	7	Planification 2ème étape de mises en œuvre LOT 2	Etudier les résultats obtenus avec POC. Tester les scénarios de pannes.
MISE EN ŒUVRE REDACTION	8		Etudier divers modules et outils pour historique ou métrologie. Installer et configurer l'outil de l'historique.
MISE EN ŒUVRE REDACTION	9		Installer et configurer l'outil de l'historique. Faire des mesures de volume de checks et charge réseau.
RECETTAGE REDACTION	10	Des ébauches de procédures de déploiement (ex : comment intégrer la surveillance d'une nouvelle application, d'un nouveau routeur etc…) V 1.0 (Release)	Rédiger le mémoire et documents technique. Finaliser la solution. Rédiger les procédures de déploiement.
RECETTAGE REDACTION MISES À JOUR	11	Des ébauches de procédures de déploiement Manuel opérationnel Les axes d'améliorations possibles	Rédiger le mémoire et documents technique. Livrer les documents à l'entreprise. Trouver des axes d'améliorations possibles.

** Tableau 4 Cahier Macro planification*

b. Backlog du produit

A partir du cahier de charge (chapitre 2.7) j'ai identifié les tâches à réaliser. Comme l'entreprise voulait avoir une première preuve pour la faisabilité du système j'ai divisé le projet en 2 lots. Le première consiste à déterminer un scénario de supervision cela veut dire sélectionner des éléments importants pour l'entreprise et mettre en place le serveur de supervision pour la partie concernée. Et le deuxième lot consiste à valider la supervision et ajouter d'autre outil afin d'améliorer la supervision.
Cette manière de planification me donne la possibilité de changer l'outil et les méthodes de supervision si je rencontre un problème sérieux lors du lot 1.
Chaque fonctionnalité comporte une description pour expliquer la tâche et une durée de réalisation pour la planification.
Ce backlog de produit a été validé par le directeur technique après la première semaine.

	Fonctionnalité	Description	Facteur	Poids	Jours	Charge (jours et)
Etude				4.00	4.00	4.00
P0-001	Etude de l'outil	Prise en main de l'outil de la supervision. Lire les documentation et comprendre son fonctionnement.	1.00	2.00	2.00	
P0-002	Etude des plugins, modules	Etudier les modules et plugins pour choisir certaines parmi eux qui peuvent être utiles pour le projet.	1.00	1.00	1.00	
P0-003	Etude de l'infrastructure du scénario	Etudier plus profondément la partie de l'infrastructure concernant le scénario. Repérer les éléments perturbants firewall etc. Identifier tous les matériels et applications.	1.00	1.00	1.00	
POC (Mise en Œuvre)				20.00	20.00	20.00
P1-001	Définir l'architecture de supervision	Protocoles de communication, stratégie de déploiement. Trouver comment déployer sur site distant chez OVH.	1.00	1.00	1.00	
P1-002	Mettre en place une VM avec Centos	Créer une VM sur hyperviseur Esxi et la configurer. Installer un Centos pour héberger le superviseur.	1.00	0.50	0.50	
P1-003	Installation des outils de test de réseau.	Installer les outils utiles pour le réseau. Nmap, wireshark, netstat etc.	1.00	0.50	0.50	
P1-004	Installation de l'outil de supervision.	Préparer l'environnement de travail et installer l'outil.	1.00	0.50	0.50	
P1-005	Configuration de l'outil.		1.00	1.00	1.00	
'P1-006	Activer, configurer base de données.	installer serveur de base de données et le configurer dans le superviseur. Activer la fonctionnalité de sauvegarde dans BD. Tester les sauvegardes.	1.00	1.00	1.00	
'P1-007	Activer, configurer interface web.	Installer les services nécessaires pour faire marcher interface web. Installer le module d'interface web et coupler avec le superviseur Configurer l'accès depuis interne et externe.	1.00	0.50	0.50	
'P1-008	Installation des plugins.	Installer les plugins et les configurer pour ajouter les éléments à surveiller. Activer ces fonctionnement dans le superviseur. Installation sur les équipements monitoré si nécessaire.	1.00	2.00	2.00	
'P1-009	Ajouter un host.	Ajouter une machine physique dans la configuration de l'outil. Paramétrer les facteurs à surveiller. Définir les seuils, les fréquences de check et les alertes.	1.00	1.00	1.00	
'P1-010	Ajouter un switch.	Ajouter l'équipement dans le superviseur. Paramétrer les facteurs à surveiller. Définir les seuils, les fréquences de check et les alertes.	1.00	0.50	0.50	
'P1-011	Ajouter routeur.	Ajouter l'équipement dans le superviseur. Paramétrer les facteurs à surveiller. Définir les seuils, les fréquences de check et les alertes.	1.00	1.00	1.00	
'P1-012	Ajouter dépendances.	Définir les dépendances des équipements ajoutés et de les ajouter dans la configurations du superviseur.	1.00	0.50	0.50	
'P1-013	Ajouter FAI.	Ajouter configuration pour tester accès internet depuis le point de sortie du routeur.	1.00	1.00	1.00	
'P1-014	Ajouter Hyperviseur Esxi OVH.	Créer compte vSphere superviseur. Ajouter configuration vSphere dans le superviseur. Ajouter l'hyperviseur dans la configuration de l'outil. Déploiement sur serveur Esxi. Paramétrer les facteurs à surveiller. Définir les seuils, les fréquences de check et les alertes. Tester la surveillance.	1.00	1.00	1.00	
'P1-015	Ajouter VM.	Ajouter VM dans la configuration de l'outil. Paramétrer les facteurs à surveiller. Définir les seuils, les fréquences de check et les alertes. Tester la surveillance.	1.00	1.00	1.00	
'P1-016	Ajouter dépendances et tester.	Ajouter les dépendances nécessaires pour la VM et hyperviseur. Tester les dépendances.	1.00	1.00	1.00	
'P1-017	Ajouter Serveur MySQL.	Créer compte MySQL pour superviseur. Configurer le plugin et ajouter serveur MySQL dan le superviseur.	1.00	1.00	1.00	
'P1-018	Ajouter Serveur HTTP.	Ajouter un serveur HTTP dans la configuration de l'outil. Paramétrer les facteurs à surveiller. Définir les seuils, les fréquences de check et les alertes.	1.00	1.00	1.00	
'P1-019	Ajouter serveur FTP	Ajouter un serveur FTP dans la configuration de l'outil. Paramétrer les facteurs à surveiller. Définir les seuils, les fréquences de check et les alertes.	1.00	1.00	1.00	
'P1-020	Ajouter modem.		1.00	1.00	1.00	
'P1-021	Manuel d'installation et configuration.	Rédactions des manuels d'installations et configurations concernant le serveur de supervision, l'outils et les plugins utilisés.	1.00	1.00	1.00	
'P1-022	Tester scénario de panne.	Simuler certains scénarios de panne. Faire tomber les équipement ou les applications de test. Ce test comprend un mal fonctionnement d'un service hébergé sur VM, panne d'une VM, problème dans Hyperviseur, problème du réseau, anomalie dans réseau et poste interne. Visualiser ces états dans interface. Surveiller les notifications envoyés. Observer les acquittions.	1.00	1.00	1.00	

Ce lot 2 était prévu au début pour étendre le système de supervision mais après la réunion de la fin du POC, nous avons décidé de le consacrer pour les tests, l'outil historique et rédactions de documents.

Lot 2 : Tests et historique				21.50	21.00	21.00	
P3-001	Etudier des outils à intégrer dans shinken	Trouver des outils utiles pour faire historique, métrologie, rapporting. Etudier la communications et intégration avec le moteur shinken.	1.00	2.00	2.00		
P3-002	Gestion de projet.	Suivi du projet, réunion et démos.	1.00	0.50	0.50		
P3-003	Faire des mesures.	Faire des mesures pour les volumes de check et charge réseau. Pour vérifier que les checks ne surcharge pas le réseau.	1.00	1.00	1.00		
'P3-004	Intégration des outils de métrologie et reporting dans shinken.	Installation des outils pour avoir un historique, pour faire la métrologie avec des graphiques.	1.00	7.00	7.00		
'P3-005	Finir d'ajouter les dépendances et des éléments de corrélations dans shinken.	Déterminer et ajouter les dépendances dans shinken. Ajouter les éléments parents. Définir le alertes par rapport aux dépendances.	1.00	1.00	0.50		
'P3-006	Tester scénario de panne.	Rédiger les cas de test. Simuler certains scénarios de panne. Faire tomber les équipement ou les applications de test. Visualiser ces états dans interface. Surveiller les notifications envoyés. Observer les acquittions. Debugger l'outil en cas de problème.	1.00	4.00	4.00		
'P3-007	Validation des éléments supervisés	Tester le superviseur pour vérifier les fonctionnalités identifiés.	1.00	1.00	1.00		
P3-008	Rédaction des ébauches de procédures de déploiement et les axes d'améliorations possibles. Fin de travail de diplôme.	Rédiger des documentations technique pour expliquer les procédures de déploiement. Comment ajouter un service ou équipement dans superviseur. Comment changer la configuration des éléments surveillées ; fréquence d'alerte, les contacts, IP d'un host etc. Comment modifier les règles de surveillance. Identifier les améliorations possibles. Mettre à jour les schémas, procédure, base de login de Qim info. rédaction du mémoire	1.00	5.00	5.00		

* *Tableau 5 Backlog du produit*

c. Rentabilité

Faible coût. Tous les outils installés sont open source et gratuits. Mais l'installation de ces outils prend beaucoup de temps.

d. Coût

- Heures de travail pour mise en œuvre
- Serveur de supervision

e. Gains

On peut estimer quelques gains mais les vrais gains, on verra une fois que le système est étendu sur l'infrastructure.

- Heures de travail de l'admin est optimisé et il sera proactif.
- Rapidité de la résolution des problèmes.
- Moins d'interventions urgentes et coûteuses.
- Identification des faiblesses et supervision sur la base de métriques.

f. Contraintes

i. Délais

Le projet débute le 20 avril et prend la fin le 8 juillet ce qui fait 11 semaines. Mais une semaine est prévue pour les mises à jour après cette date.

ii. Technique

- Etre facile d'utilisation
- Etre facile à paramétrer
- Envoyer des mails en cas d'alerte

- Etre compatible avec VMware
- Etre compatible avec système distribué pour des sites dispersé géographiquement
- Etre compatible avec système hétérogène
- Les éléments à surveiller peuvent varier ou l'infrastructure peut subir des modifications. La solution doit s'adapter à l'infrastructure. Une souplesse est nécessaire pour ajouter des modules, pour faire des modifications.

g. Livrables

- Le système de supervision opérationnel
- Schéma de l 'architecture du monitoring
- Une étude sur les différentes solutions de supervision libre
- Un manuel opérationnel (utilisateur)
- Un manuel d'installation de la solution
- Un manuel de paramétrage de la solution
- Les axes d'améliorations possibles
- (option) Des ébauches de procédures de déploiement (ex : comment intégrer la surveillance d'une nouvelle application, d'un nouveau routeur etc…)

h. Critères d'acceptations

La solution fournie doit remplir certaines conditions pour être une solution acceptable.
Supporter infrastructure distribué et hétérogène
Doit donner la possibilité de la déployer. (Ex. : ajouter un parc, un serveur …)
Le système doit détecter la panne en prenant en compte les dépendances. (Ex. : switch en panne et il ne doit pas dire tous ce qui liés à ce switch sont en panne)
Alerter au bon moment et dans certaines limites de temps et degré d'importance

i. EQUIPE

Le projet sera réalisé par Huseyin et surveillé par le directeur technique de Qim info M. DAUDEL et M. Hostettler qui est ingénieur de système en tant que collaborateur externe. Le chef de projet M. Loïc est là pour le bon déroulement du projet.

j. Recettage

La solution de supervision installée doit être testée afin de vérifier les spécifications indiquées dans le chapitre 10.4.1 et dans le cahier de charges 4.2 .

Un tableau de test préparé montre les tests à effectuer avant la sortie d'une version Bêta de la solution.

k. Risques

J'ai commencé à déterminer les risques à partir de ce que je devais réaliser et je devais toucher. Les retours des autres réalisations m'ont aidé pour identifier les risques aussi. Avec les projets qui touchent l'infrastructure, nous avons toujours certaines risques comme ; d'être bloqué par les équipements ou applications, provoquer certaines pannes ou charger le réseau. Comme il y a un système d'envoie mail avec la source de panne, cela risque des envois excises et avec des fausses alertes. Le choix d'outil peut aussi être un risque car en cas de d'un mauvais choix j'ai un risque. Un pour un projet informatique peut facilement comporter le risque de dépassement compétences s'il est mené par un débutant dans les domaines.
A partir de tous ces risques j'ai établi ma carte de risque en identifiant aussi son impact, son type, ses conséquences mais aussi les actions préventives à mener. J'ai déterminé des dates pour fermer les risques par rapport aux étapes des réalisations.
Tous les risques doivent être fermés au plus tard après les tests. Cette gestion de risque est suivie par le consultant William Ray et les fermetures de risques sont confirmées par lui.

Description du risque : description textuelle des facteurs de risque ainsi que de leur contexte d'apparition. Si de nouveaux éléments apparaissent venant compléter le contexte d'apparition du risque, ils sont ajoutés au fur et à mesure dans cette case, précédés par la date de mise à jour
Poids : poids est calculé avec la multiplication de l'impact et de la probabilité d'un risque.
Actions préventives : actions engagées ou à engager dans le but de réduire le risque.

Date	Description du risque	Impacts	Type	Proba bilité	Impa ct	Poids	Pour le	Actions préventives
27.04	Les configurations des outils de sécurité (ex. : firewall, anti-virus, architecture virtuelle) ou certaines équipements (routeur, modem) peuvent empêcher certaines communications du système de surveillance sans être repéré.	Charge de travail (délais, coût formation)	Technique	3	3	9	8.5.15	Identifier les éléments bloquants comme firewall et ajouter les régles.Demande l'aide de Grégoire.
27.04	Administrateurs peuvent être dépassés suite aux nombres de messages reçus.	Les administrateurs ignorent les messages	Technique	3	4	12	14.5.15	Réduire le nombre de notifications. Limiter le nombre d'objet surveillé en définissant l'architecture d'envoi des alertes.
27.04	Problème de dépendances, le système peut se tromper pour la source de panne. Fausses alertes.		Technique	2	3	6	14.5.15	Bien étudier le système et rétablir tous les dépendances sur le superviseur. Bien comprendre le fonctionnement de l'architecture technique de Qim info.
27.04	Compétences collaborateur	Charge de travail (délais+coût formation)	Technique	2	3	6	14.5.15	Collaborer avec Grégoire.
27.04	Saturation du réseau à cause de volume de flux généré.		Technique	2	4	8	14.5.15	Limiter la fréquence de test par le superviseur.
27.04	Superviser les éléments inutilement.	Charge de travail (délais+coût formation)	Fonctionnel	2	1	2		Définir les éléments **principaux** à surveiller par rapport à leurs importances.
27.04	Compréhension fonctionnelle	Développements inadéquat	Fonctionnel	1	3	3		Avoir une documentation technique et fonctionnelle
27.04	Un test va perturber le fonctionnement normal de la ressource testée à cause d'un effet de bord.	Répercussions sur le Business As Usual.	Organisation	1	4	4		Contrôler le fonctionnement normal des appareils lors des tests.

** Tableau 6 Cahier de Risques*

Annexe B. Lexiques

- BAM (Business Activity Monitoring): Supervision de l'activé métier.
- Métrologie : La science de la mesure. C'est donc l'ensemble des techniques et des savoir-faire permettant de donner une valeur à une observation, en bref de mesurer.
- OVH : OVH est un hébergeur de sites web français. Avec environ 180 000 serveurs en octobre 2014, OVH dispose de l'un des plus grands parcs de serveurs au monde.
- ESXI MRTG (Multi Router Traffic Grapher) : Logiciel gratuit basé sur SNMP permettant la traduction graphique de données
- MIB (Management Information Base), Base d'informations structurée d'un équipement utilisé par SNMP
- NRPE (Nagios Remote Plugin Executor) : Module Nagios pour exécuter des plugins sur une machine à distance.
- FORK : Un nouveau logiciel crée à partir du code source d'on logiciel existant.
- POSIX : Une famille de normes techniques en informatique définie depuis 1988 per IEEE.
- POC (Proof of concept) : Démonstration de faisabilité, une réalisation d'une méthode pour démontrer sa faisabilité.
- SNMP (Simple network Management Protocol): Protocole de communication pour gérer les équipements à travers le réseau.
- OID (Object Identifier) : Nom définissant un objet dans le MIB de SNMP.
- WMI (*Windows Management Instrumentation) :* Gestion interne de surveillance de Windows.

Annexe C. Liens Utiles

1-Business monitoring : http://fr.wikipedia.org/wiki/Business_activity_monitoring

2- Doc officiel Shinken - http://shinken.readthedocs.org

3- Forum Shinken en anglais: http://forum.shinken-monitoring.org/

4- Forum Monitoring : http://forums.monitoring-fr.org/

5- Travaux Shinken à hepia : http://www.tdeig.ch/shinken/

6-Limiter les alertes : http://connect.ed-diamond.com/GNU-Linux-Magazine/GLMF-137/Ne-pas-devenir-dingue-avec-sa-supervision

7- Interview avec Jean Gabès- http://blog.nicolargo.com/2012/09/interview-de-jean-gabes-pour-la-sortie-de-shinken-1-2.html

8- Introduction au Shinken 2.2 : http://shinkenlab.io/release-2-2/

9 -Plugins pour monitoring : https://www.monitoring-plugins.org/

Annexe D. Procédure d'installation Hyperviseur Esxi 5.5 et VSphere Client

a. Téléchargement de l'image depuis WMWare

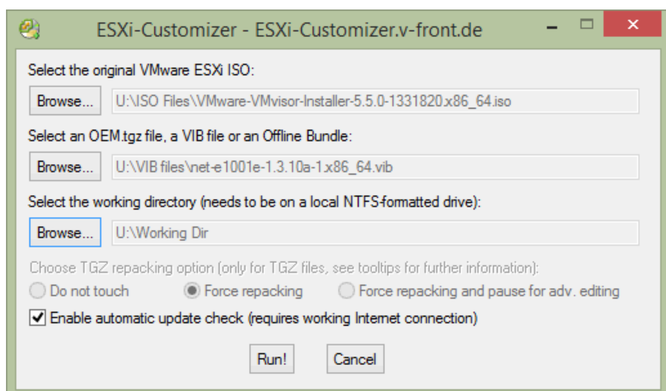
Créer un compte sur le site Wmware.com pour pouvoir télécharger l'image et avoir une licence gratuite. On peut choisir la version d'Esxi grâce au comparateur mise en disposition. Il faut choisir la version par rapport au matériel, l'OS de la VM à installer. Le matériel à disposition support Esxi 5.5 mais pas 6.0.

Guide Esxi http://www.vmware.com/resources/compatibility/search.php

Pour installer et utiliser Esxi 5.5, vos ressources matérielles et système doivent répondre aux exigences suivantes :

- Esxi 5.5 n'installera et n'exécutera que les serveurs dotés de CPU x86 64 bits.
- Esxi 5.5 nécessite une machine hôte disposant d'au moins deux cœurs.
- Esxi 5.5 ne prend en charge que les instructions de CPU LAHF et SAHF.
- Esxi 5.5 nécessite d'activer le bit NX/XD pour le processeur dans le BIOS.

Les cartes réseaux de Realtek R8168/8169 ne sont pas prises en charges mais la modification de l'image Esxi est possible. Pour modifier une image télécharger le logiciel Esxi customizer et les pilotes pour les cartes réseaux sur le même site.

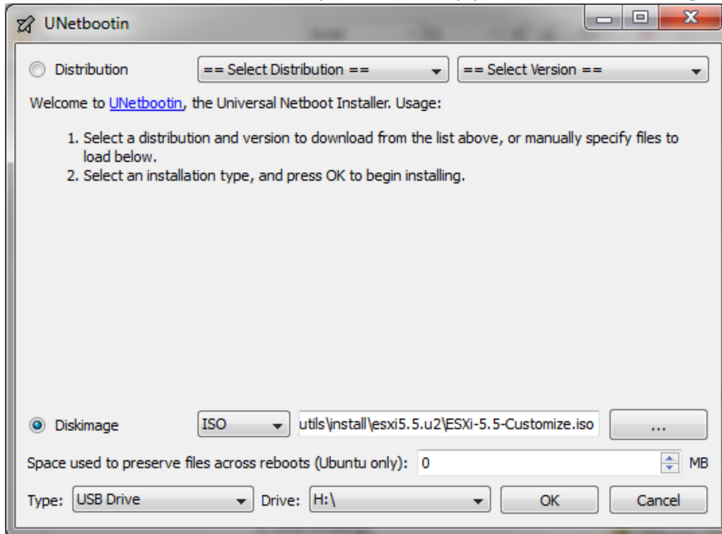


Esxi customizer : http://www.v-front.de/p/esxi-customizer.html

Lancer l'Esxi customizer après l'installation. Sélectionner l'image Esxi et le pilote à ajouter comme le

* *Figure 67 Modifier l'image Esxi*

b. Préparer le support de démarrage



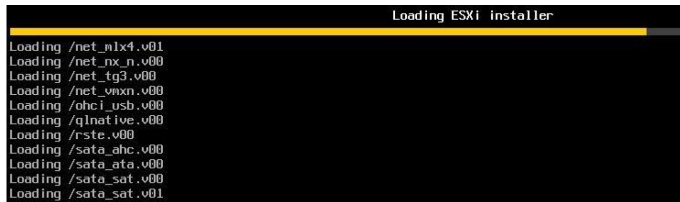
Télécharger le logiciel unetbootin pour préparer une clé de démarrage avec Esxi.

Site de téléchargement : http://unetbootin.sourceforge.net/

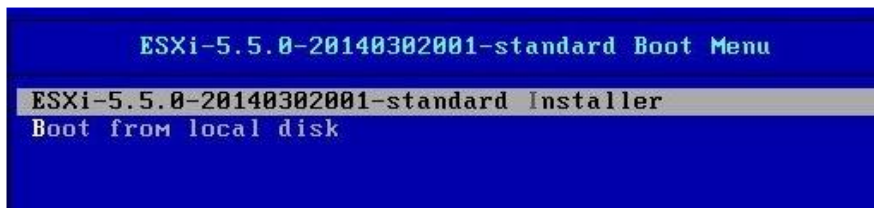
Lancer le et sélectionner l'image et la clé USB.

* *Figure 68 Préparer clé USB de démarrage*

c. Installation Esxi



* *Figure 69 Chargement des composants de l'Esxi*



* *Figure 70 Menu d'installation Esxi*

Démarrer l'ordinateur sur la clé USB et sélectionner l'installation Esxi dans le menu qui est affiché.



* *Figure 71 Choisir mot de passe root*



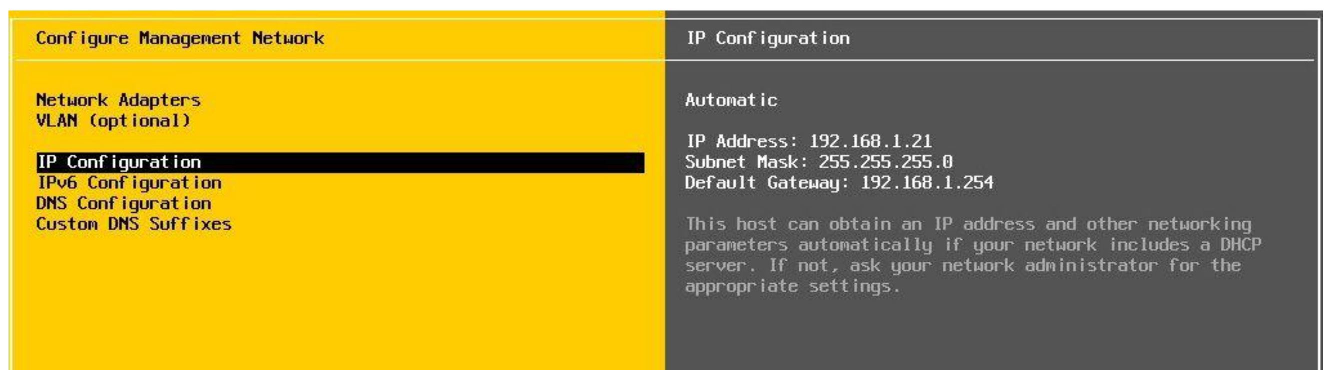
* Figure 72 Fin de l'installation Esxi

Il faut démarrer le PC et configurer la carte réseau avec une IP fixe. Appuyer sur F2 et entrer le mot de passe du root.



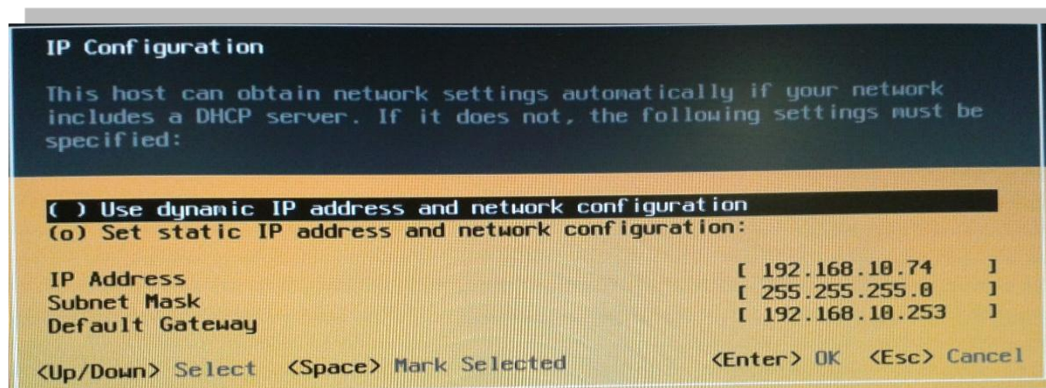
* Figure 73 Menu de configuration d'Esxi

La machine se configure via le DHCP au démarrage et il faut modifier pour un avoir une IP fixe.



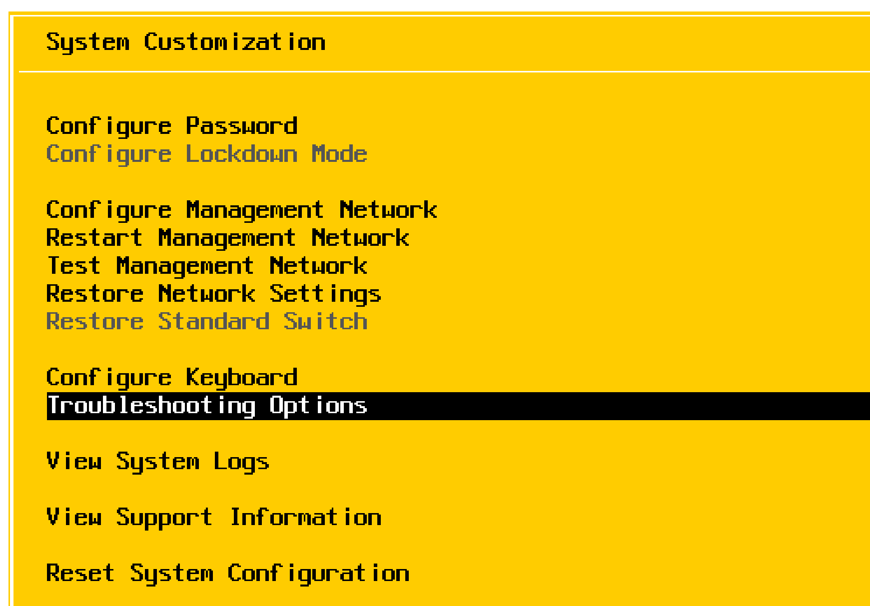
* Figure 74 Menu Configuration de l'IP

Choisir configuration manuel, entrer l'IP du serveur DNS, choisir un nom et une IP disponible.



* *Figure 75 Configuration IP*

Il faut aussi activer le SSH sur machine Esxi pour la gestion via VSphere.



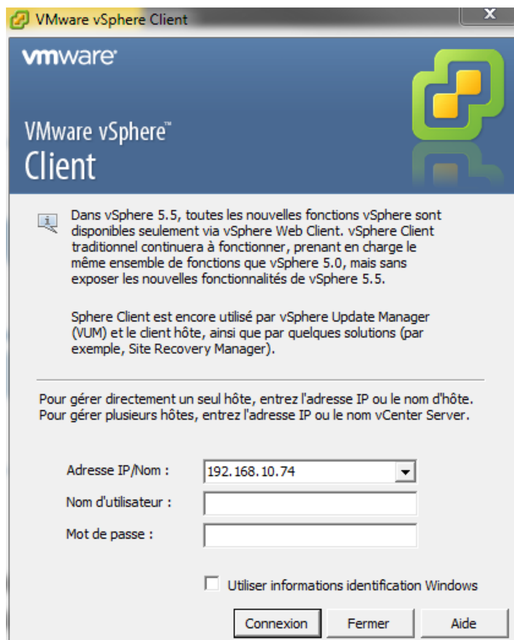
* *Figure 76 Menu TroubleShooting*



* *Figure 77 Activer SSH*

Le hyperviseur Esxi 5.5 comporte une vulnérabilité lié OpenSSL et Heartbleed, il faut appliquer les corrections KB2076121 et KB2076589. La procédure est expliquée sur le site de WMWare.
http://kb.vmware.com/selfservice/search.do?cmd=displayKC&docType=kc&docTypeID=DT_KB_1_1&externalId=2076692

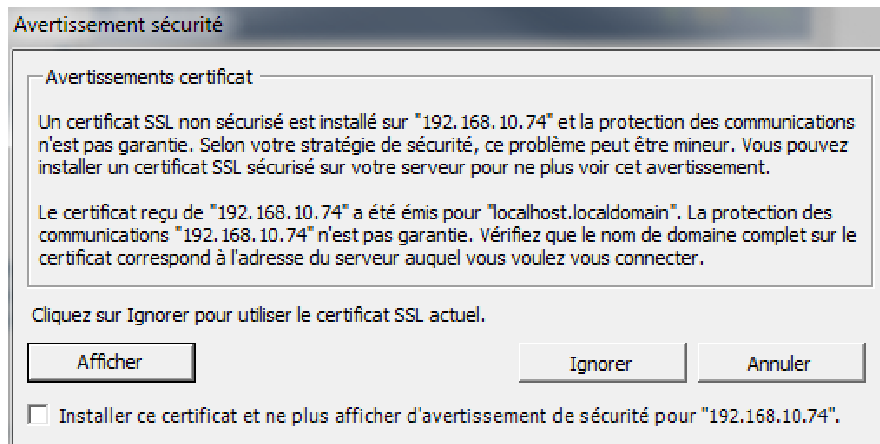
Annexe E. Procédure d'installation vSphere



Ouvrir un navigateur et taper l'IP de la machine Esxi et la page web affiché contient un lien pour télécharger le client vSphere. Après l'installation de l'exécutable lancer le logiciel vSphere. Il faut entrer l'IP du serveur avec login root et mot de passe choisi sur la machine pendant l'installation.

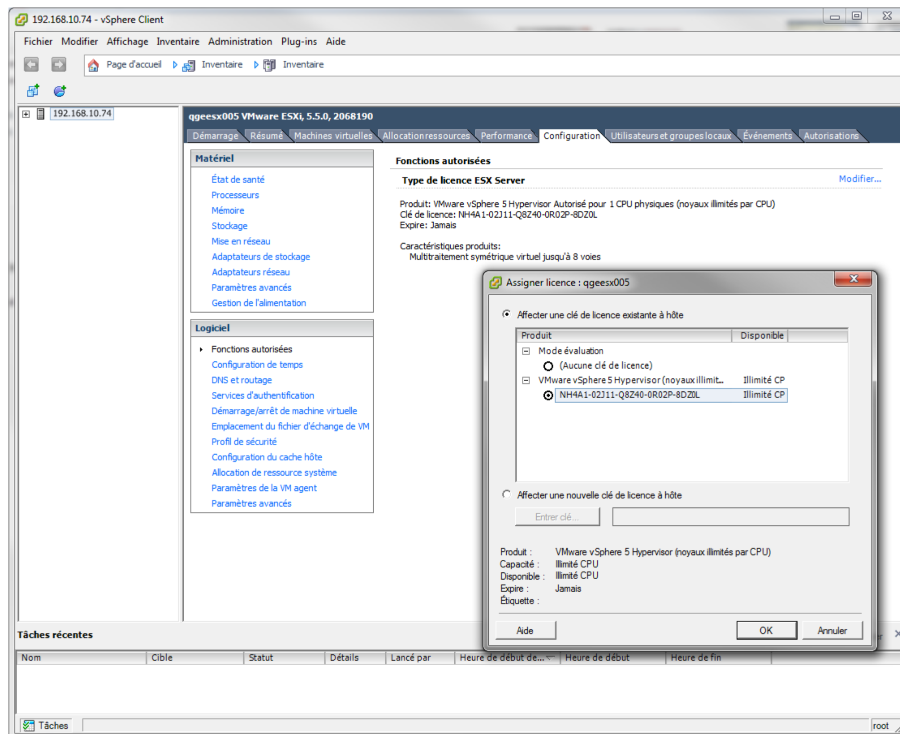
* *Figure 78 Login VSphere*

Pour la première connexion il faut soit installer le certificat de la machine soit ignorer pour se connecter.



* *Figure 79 Message concernant certificat de la machine*

La dernière étape consiste à entrer la clé de la licence obtenue sur le site WMWare.com. Elle est gratuite et affiché dans l'onglet de téléchargement de votre compte WMWare. Sinon un avertissement (Votre licence expire dans 60 jours et vos machines virtuelles seront désactivées)
Il faut sélectionner la machine dans VSphere et aller dans :
Configuration -> Fonctions autorisées -> Modifier -> Affecter une clé -> Entrer clé et entrer la clé que vous avec pris.



* *Figure 80 Entrer la clé de la licence*

Annexe F. Procédure de création VM via client vSphere

Dans menu fichier -> nouveau et machine virtuelle.

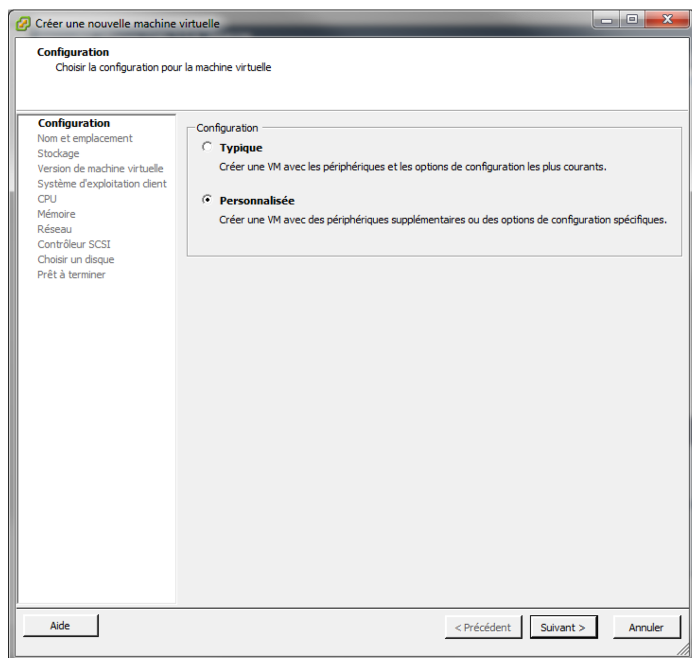


Figure 81 configurer VM personnalisée

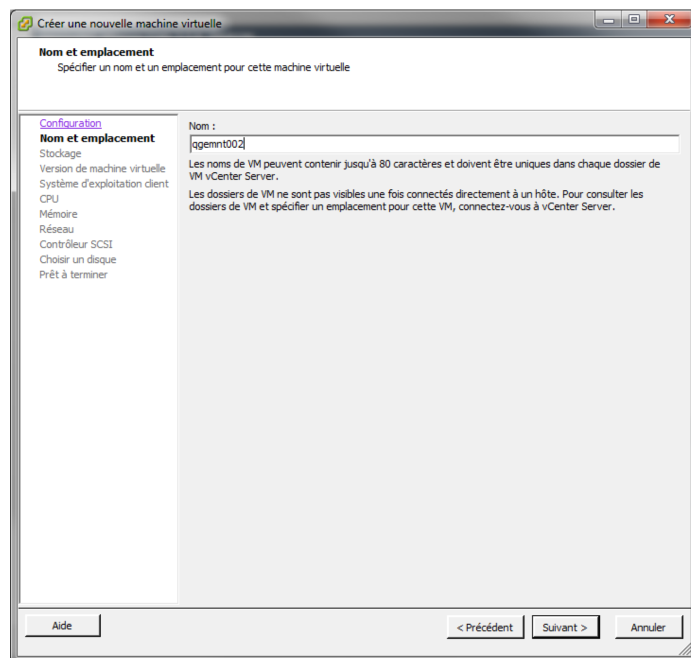


Figure 82 Sélectionner le nom de la VM

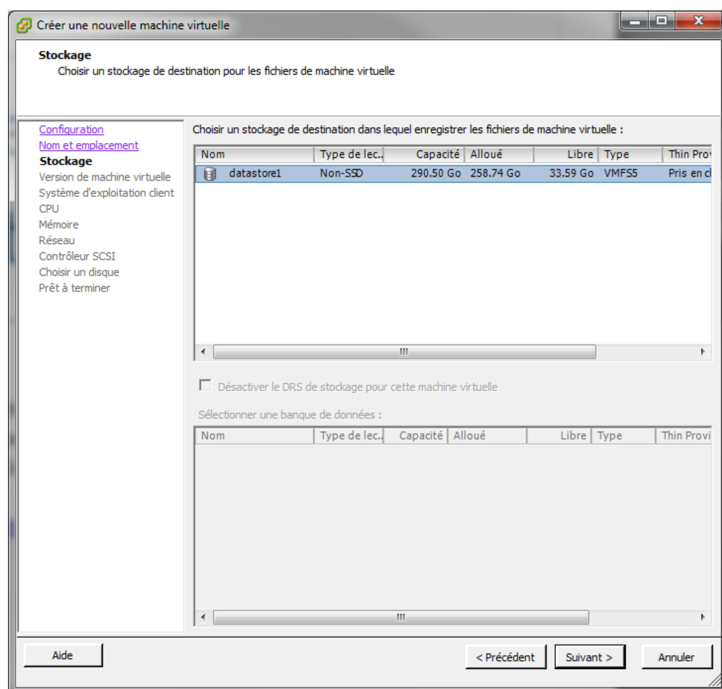


Figure 83 Choisir stockage

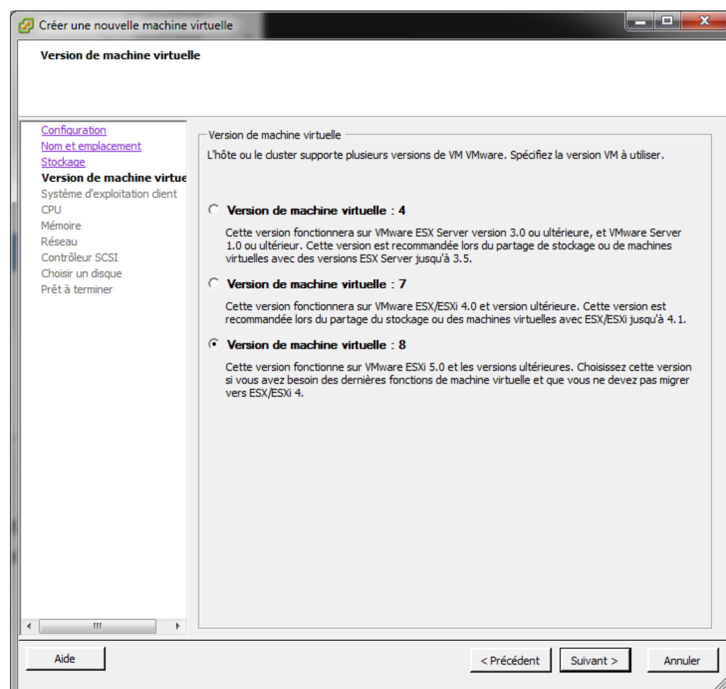
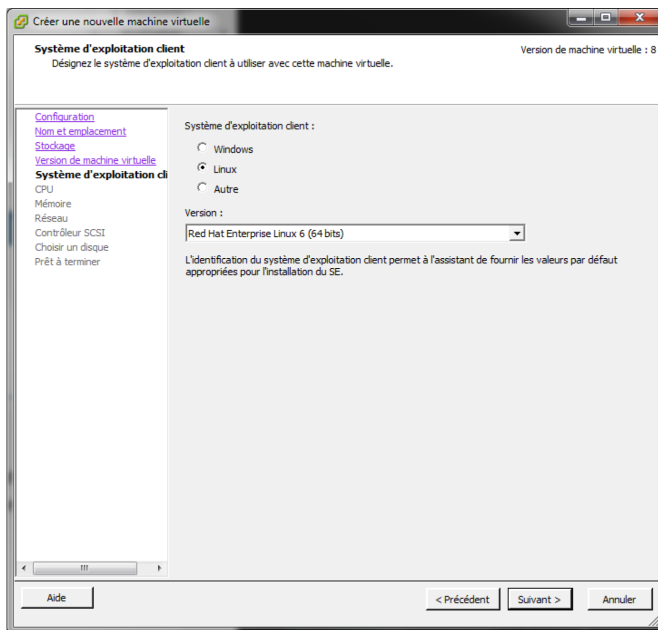
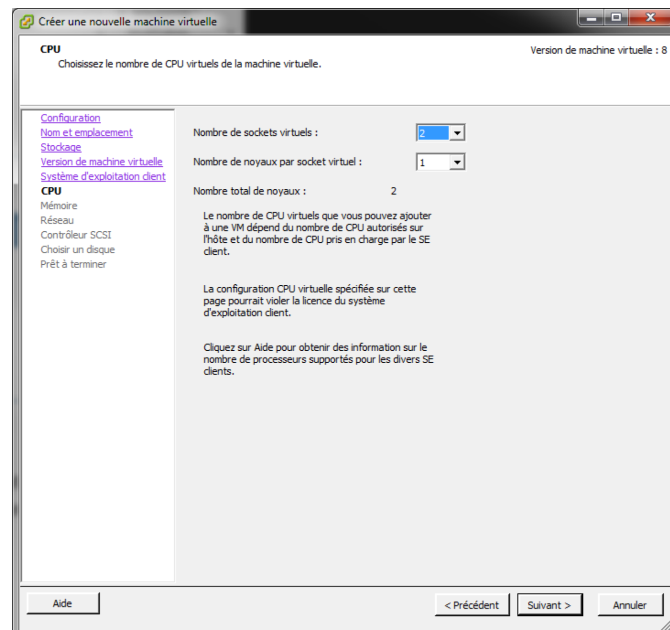


Figure 84 Choisir la version de la VM

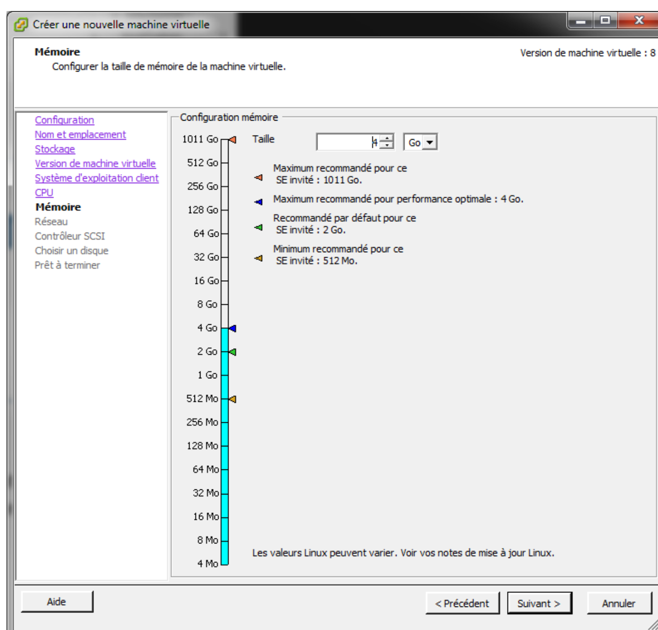


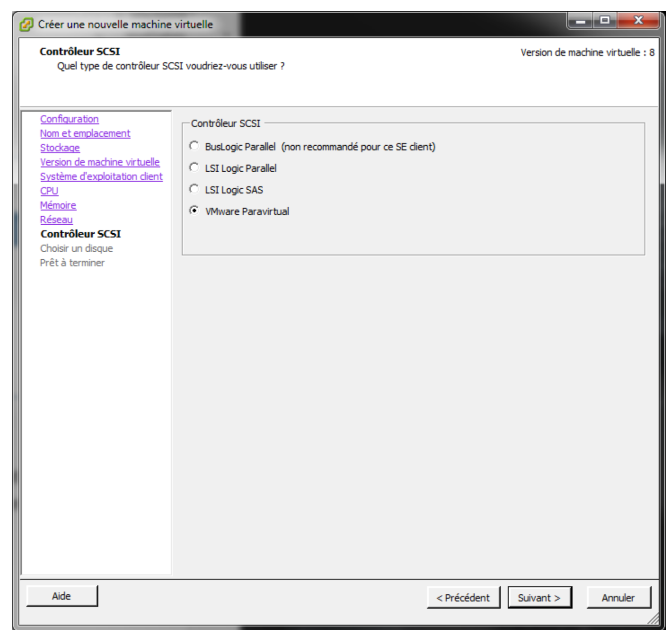


* Note : Il faut sélectionner Red Hat au lieu de Centos pour pouvoir installer différents SCSI connecteurs virtuels.

* Figure 86 Choisir le nombre sockets virtuels pour CPU

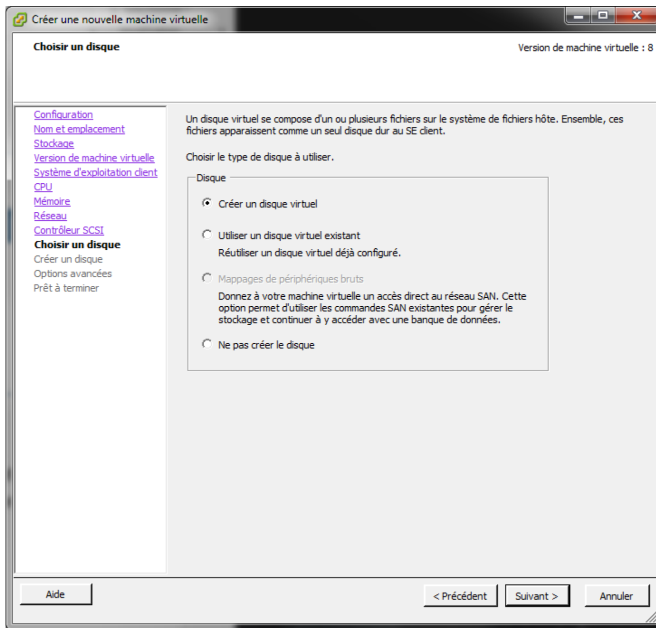
* Figure 85 Version de l'OS à installer



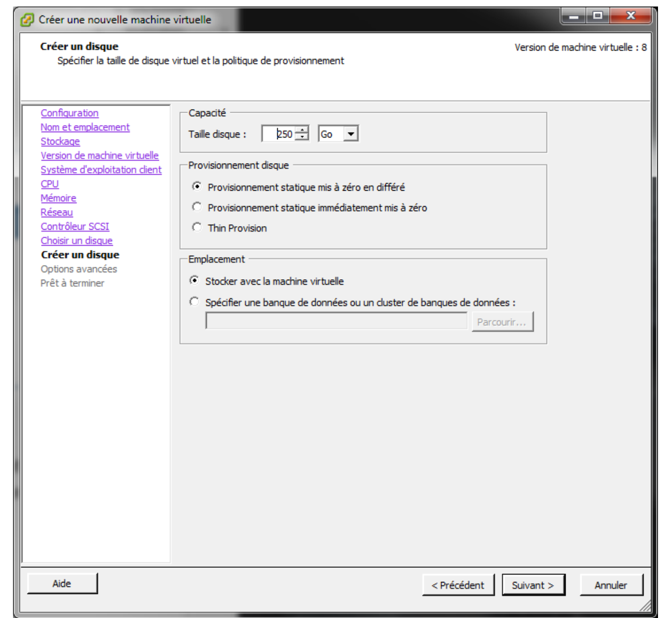


* Figure 87 Configurer la mémoire

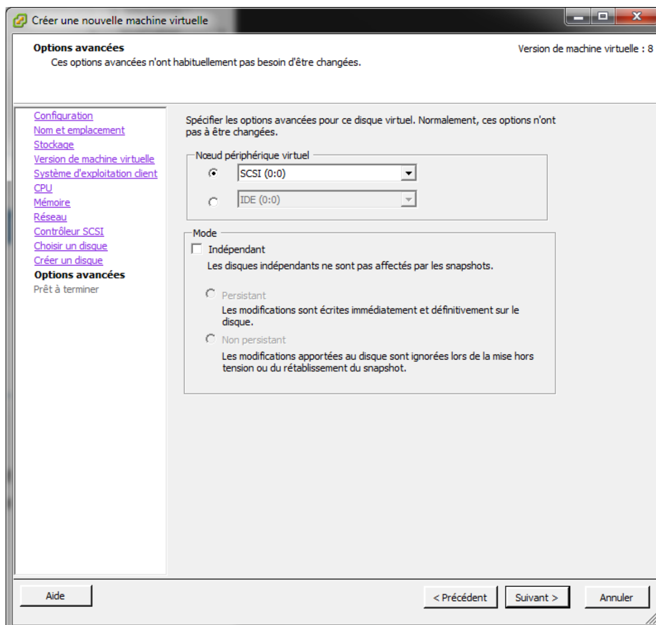
* Figure 88 Contrôleur SCSI



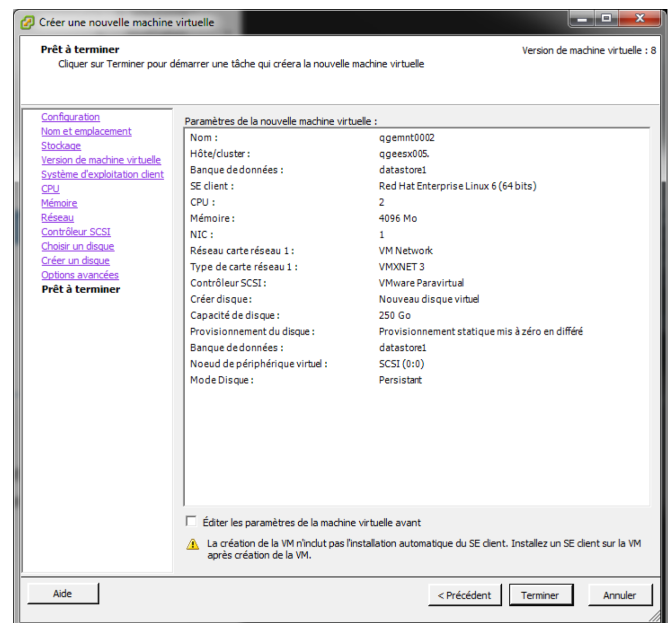
** Figure 89 Type de disque*



** Figure 90 Taille du disque*



** Figure 91 Options avancées pour le disque virtuel*



** Figure 92 Fin de la création de la VM*

Annexe G. Procédure d'installation Centos sur une VM via VSphere

a. Téléchargement de l'image Centos

Il faut tester la comptabilité de l'OS avant d'installer sur VM. Centos 7.X est supporté par Esxi 5.5 update2.

OS Vendor	OS Release	Bits	Supported Releases				
Debian	Debian GNU/Linux 7.X	32	ESXi	6.0	5.5 U2	5.5 U1	5.5
Debian	Debian GNU/Linux 7.X	64	ESXi	6.0	5.5 U2	5.5 U1	5.5

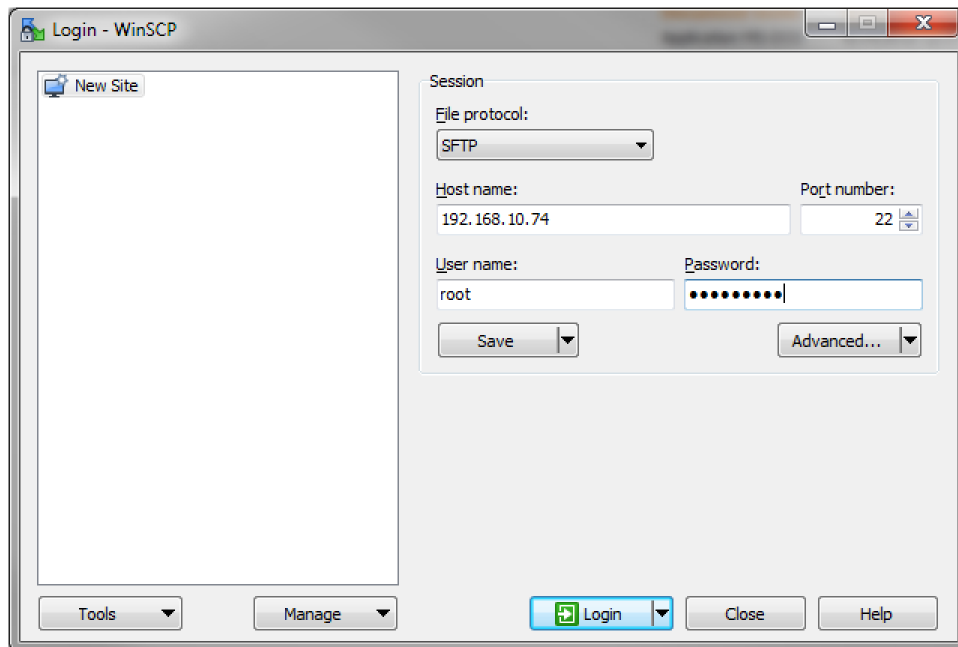
** Figure 93 Esxi compatibilités OS*

Télécharger Centos 7.1 depuis le site :
http://isoredirect.centos.org/centos/7/isos/x86_64/CentOS-7-x86_64-Minimal-1503-01.iso

b. Ajouter l'image dans la machine Esxi

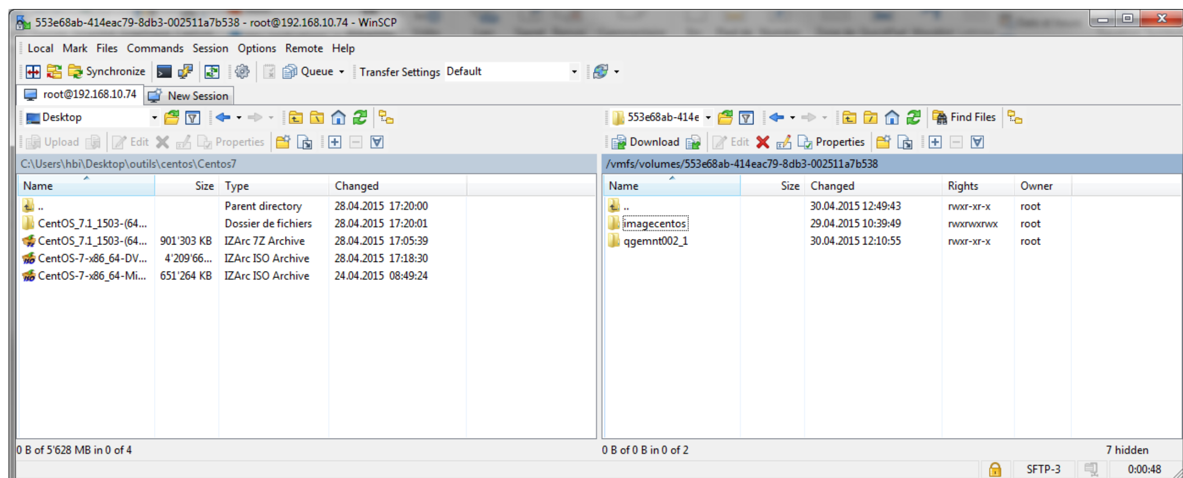
Pour pouvoir installer un OS vis vSphere on peut mettre le cd/dvd d'installation dans le lecteur de la machine ou transférer le fichier via SCP ou ftp vers la machine. Vu que le SSH est déjà activé sur Esxi après l'installation, on peut envoyer l'image via SCP avec WinSCP. En cas d'envoi par FTP, il faut activer FTP sur la machine Esxi.

Télécharger WinSCP depuis http://winscp.net/eng/download.php et lancer le avec login de la machine Esxi.

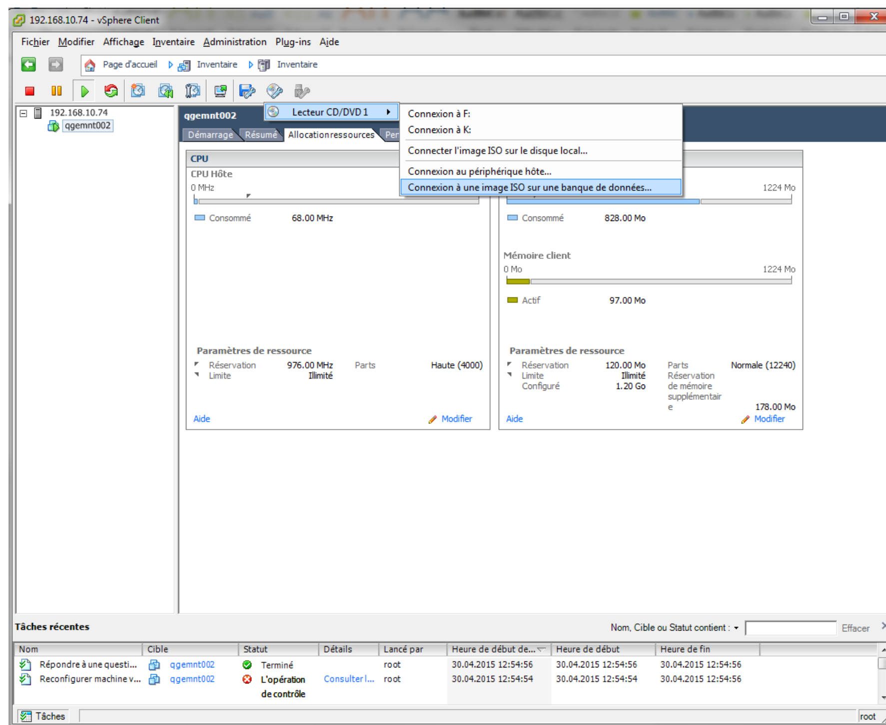


** Figure 94 Connexion avec machine Esxi via WinSCP*

Sélectionner l'image depuis local et mettre vers la machine dans l'onglet droit.

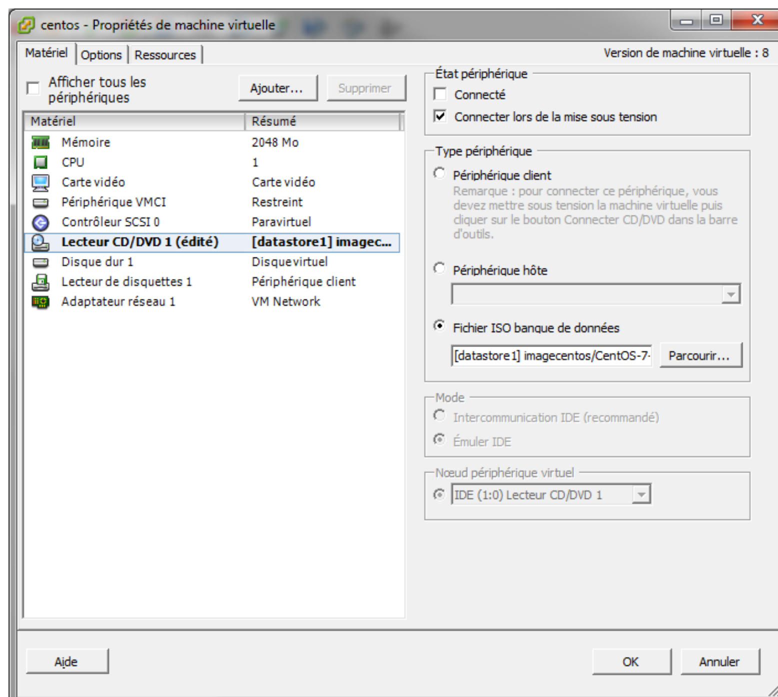


** Figure 95 Sélectionner le lecteur et connecter avec image de banque et choisir l'emplacement de l'image transféré*

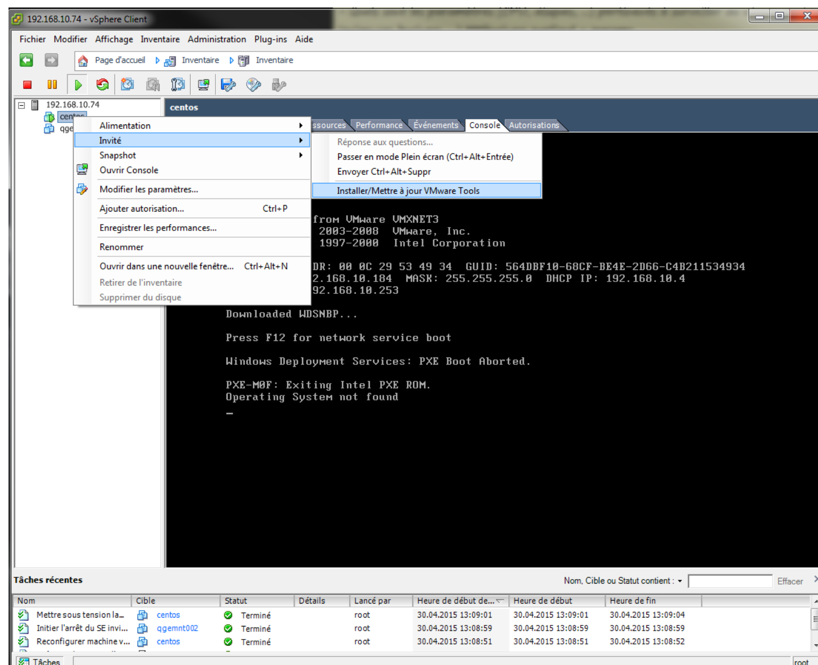


** Figure 96 Connecter l'image avec VM pour l'installer*

Sans connecter le lecteur à la VM, VM ne peut pas démarrer sur l'image.

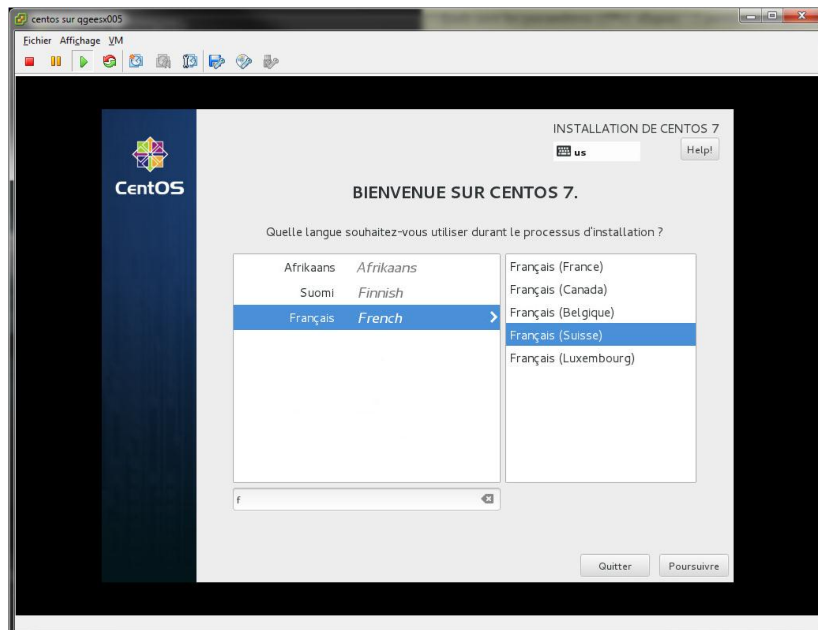


** Figure 97 Connecter le lecteur CD/DVD à la VM*



** Figure 98 installer Wm tools pour pourvoir gérer la VM correctement avec vSphere*

Maintenant on peut démarrer la VM et elle démarre sur l'image et on peut lancer l'installation.



** Figure 99 Sélectionner la langue durant installation*

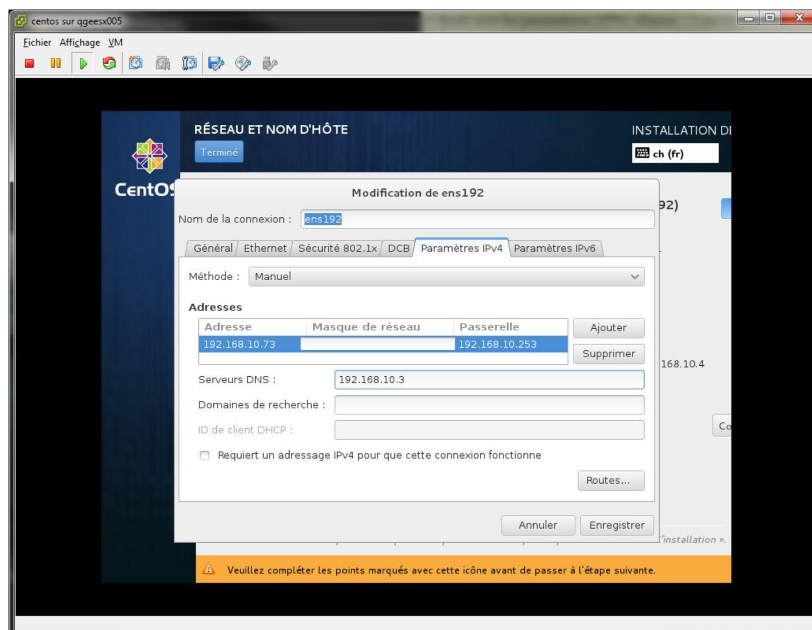


Figure 100 Configurer l'IP et réseau manuellement

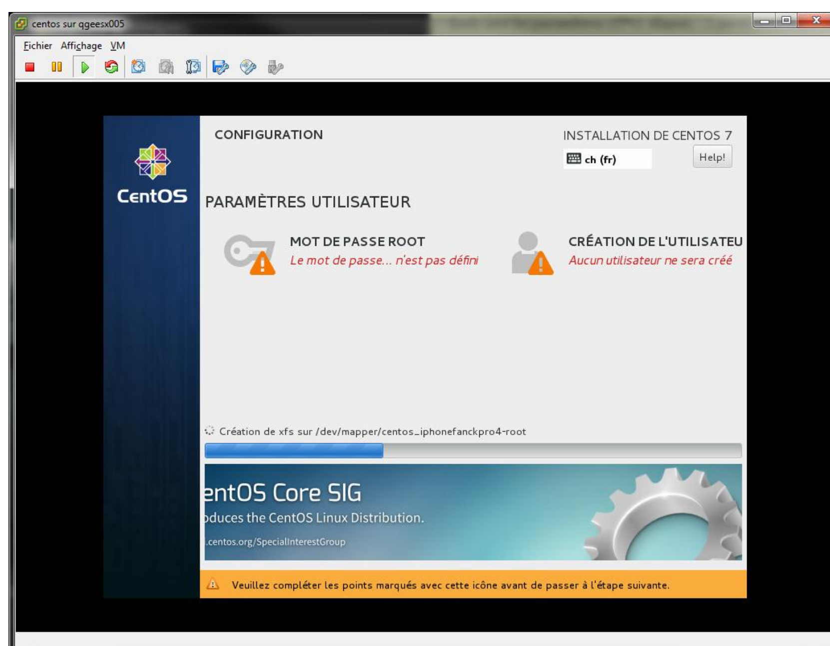


Figure 101 Sélectionner mot de passe root

L'installation minimum ne contient pas l'interface d'utilisateur. Il faut installer manuellement.
[root@qgemnt001]# yum groupinstall "GNOME Desktop" "Graphical Administration Tools"
Pour démarrer toujours avec l'interface :
[root@qgemnt001]# #ln -sf /lib/systemd/system/runlevel5.target /etc/systemd/system/default.target
Au premier démarrage une erreur affiché mais l'OS démarre.
Message : Assuming drive cache: write through
Cause : L'interface de login.
Solution : Il faut changer l'interface de login dans grub. Dans le menu de grub appuyer sur 'e' et modifier le script. Enlever le mot « **rhgb** ».

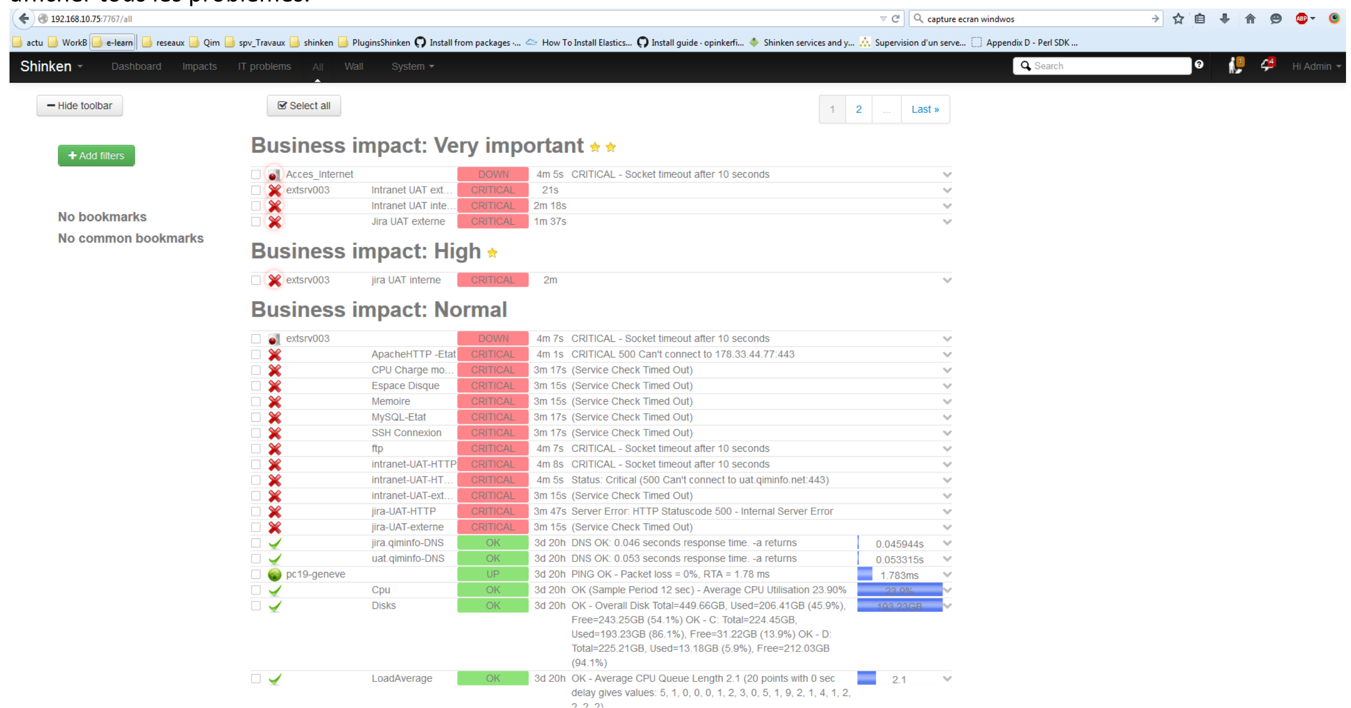
L'installation est finie.

Annexe H. Capture des tests

Suite aux tests j'ai pris de nombreuses captures d'écrans. Je mets quelques écrans concernant les accès internet, poste de Windows (PC19) et serveur extsrv003 hébergé chez OVH.

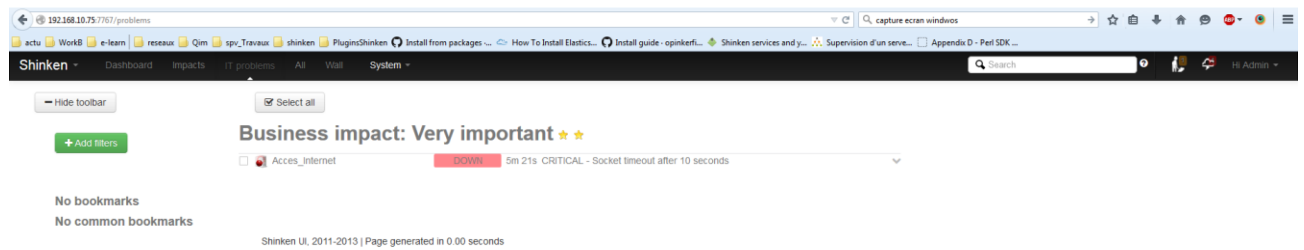
a. Coupure accès internet

En cas de coupure accès internet le Shinken ne va pas notifie les éléments dépendants. Sous la **vue All** il va afficher tous les problèmes.



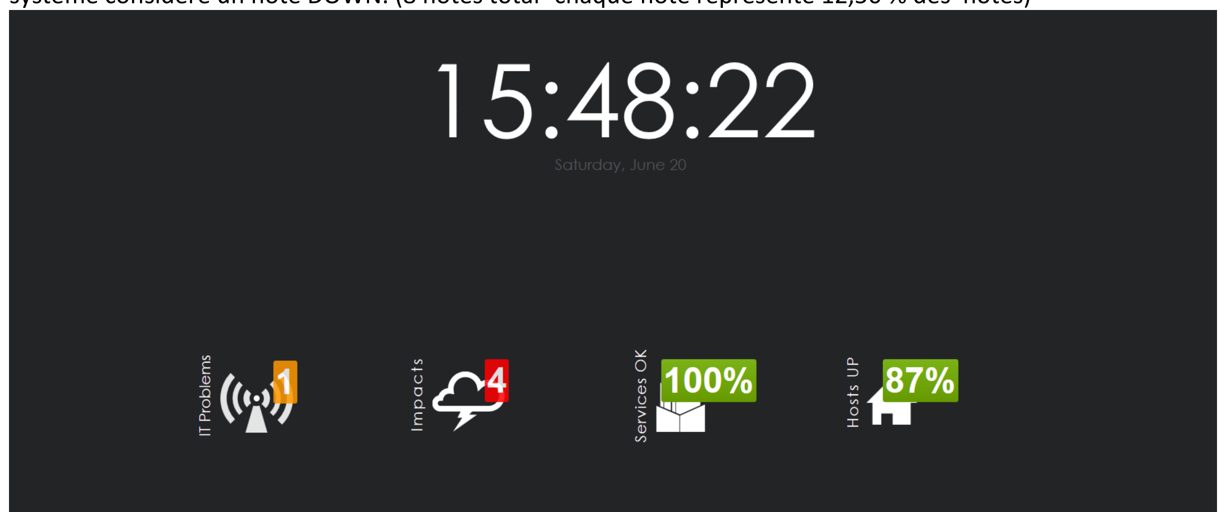
** Figure 102 Vue All en cas de coupure internet*

Mais dans la **vue IT problems** le Shinken affiche juste l'éléments perturbateur car il peut faire la corrélation de données grâce aux dépendances.



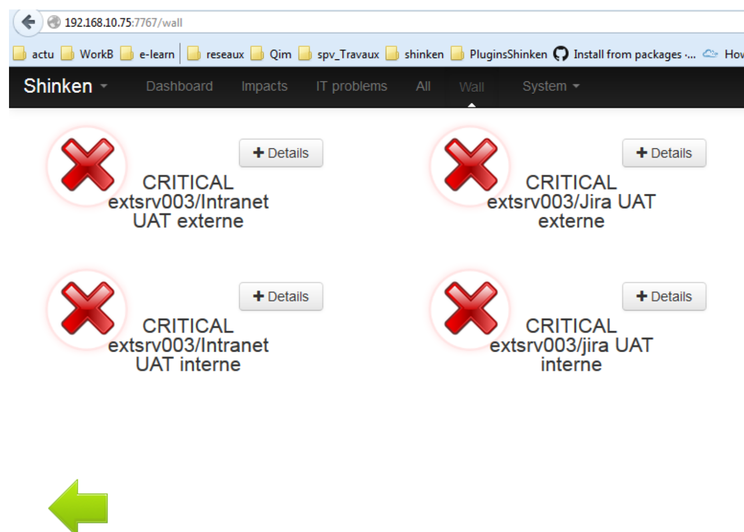
* *Figure 103 Vue IT problems pour coupure internet*

La vue Currently (Vue globale) affiche un résumé du système. Les impacts sont les règles métier touchées par le problèmes. IT problème affiche un seul problème car il ne peut pas déterminer l'état de l'infrstructure OVH qui dépend de l'accès internet. L'accès internet est déclaré comme hôte donc le système considère un hôte DOWN. (8 hôtes total- chaque hôte represente 12,50 % des hôtes)



* *Figure 104 Vue Currently*

La vue Wall affiche les impacts de business et les IT problems dans une seule vue.



There are 1 new IT problems in the last 10 minutes:

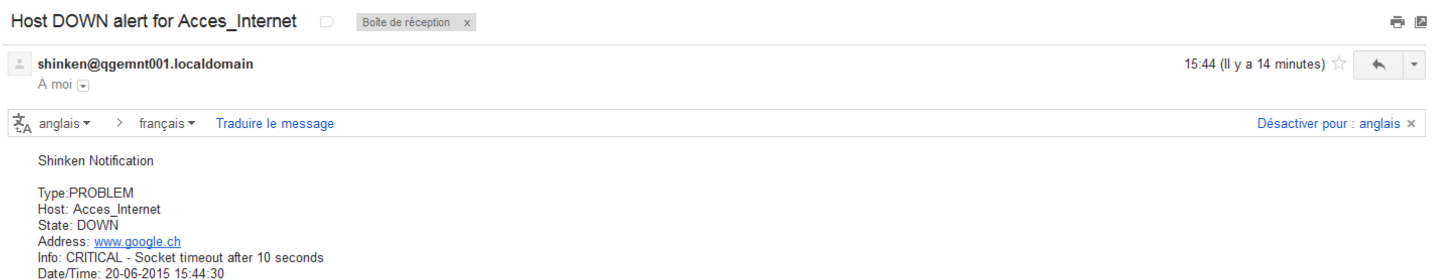
Acces_Internet is DOWN since 6m 47s

** Figure 105 vue Wall pour coupure internet*

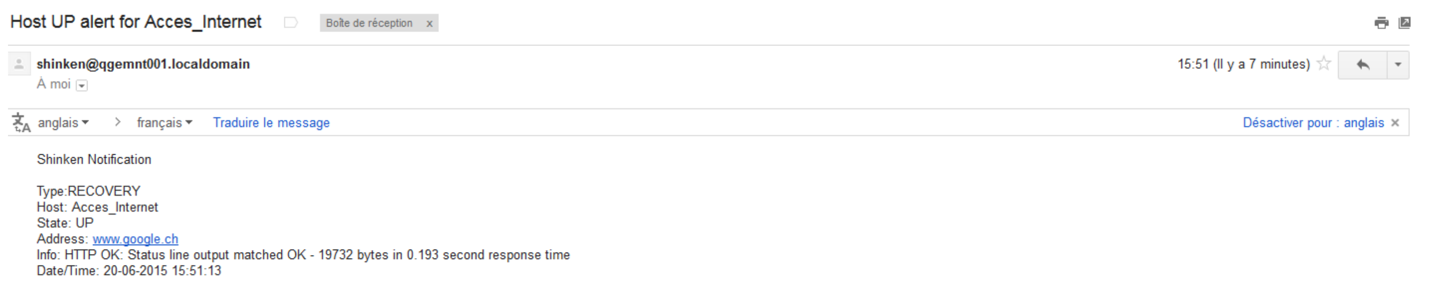
Shinken génere une seule notification pour l'accès internet et il ne vas notifier tous les éléments dépendants comme le serveur extsrv003 chez OVH.

shinken	**Host UP alert for Acces_Internet** - Shinken Notification Type:RECOVERY Host: Acces_Internet State: UP Address: www.google.ch Info: HTTP
shinken	**Host DOWN alert for Acces_Internet** - Shinken Notification Type:PROBLEM Host: Acces_Internet State: DOWN Address: www.google.ch Info:

** Figure 106 Mails reçus pour la coupure internet*



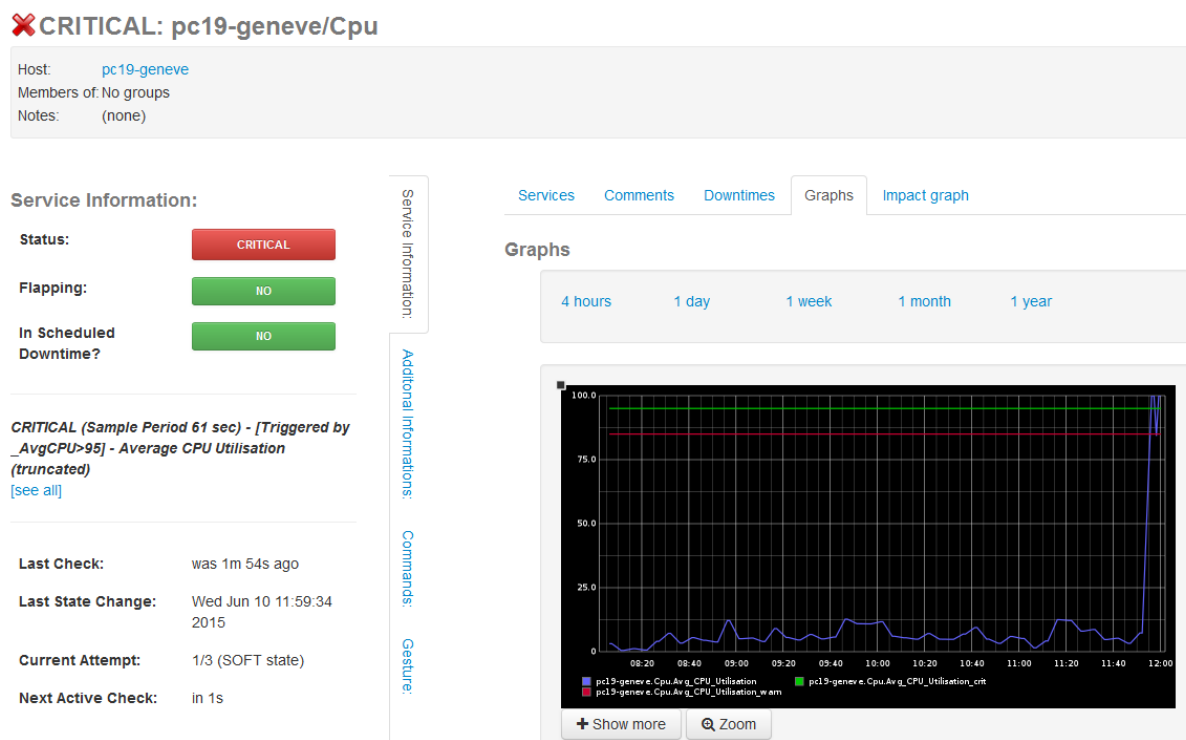
** Figure 107 Mail host down accès internet*



** Figure 108 Mail host Up accès internet*

b. Charge CPU du PC 19

Vue détaillée de la charge CPU du PC19.



** Figure 109 Etat charge CPU du pc19 sur interface web Shinken*

Mail recu suite à l'état critique de la charge CPU.

** PROBLEM alert - pc19-geneve/Cpu is CRITICAL ** Boîte de réception x

shinken@qgemnt001.localdomain
À moi

anglais > français Traduire le message

Shinken Notification

Notification Type: PROBLEM

Service: Cpu
Host: pc19-geneve
Address: 192.168.10.99
State: CRITICAL

Date/Time: 10-06-2015 12:30:06
Additional Info : CRITICAL (Sample Period 14 sec) - [Triggered by _AvgCPU>95] - Average CPU Utilisation 100.00%

** Figure 110 Mail reçu pour problème de la charge cpu du PC 19*

La notification recue suite à la résolution du problème.

** RECOVERY alert - pc19-geneve/Cpu is OK ** Boîte de réception x

shinken@qgemnt001.localdomain
À moi

anglais > français Traduire le message

Shinken Notification

Notification Type: RECOVERY

Service: Cpu
Host: pc19-geneve
Address: 192.168.10.99
State: OK

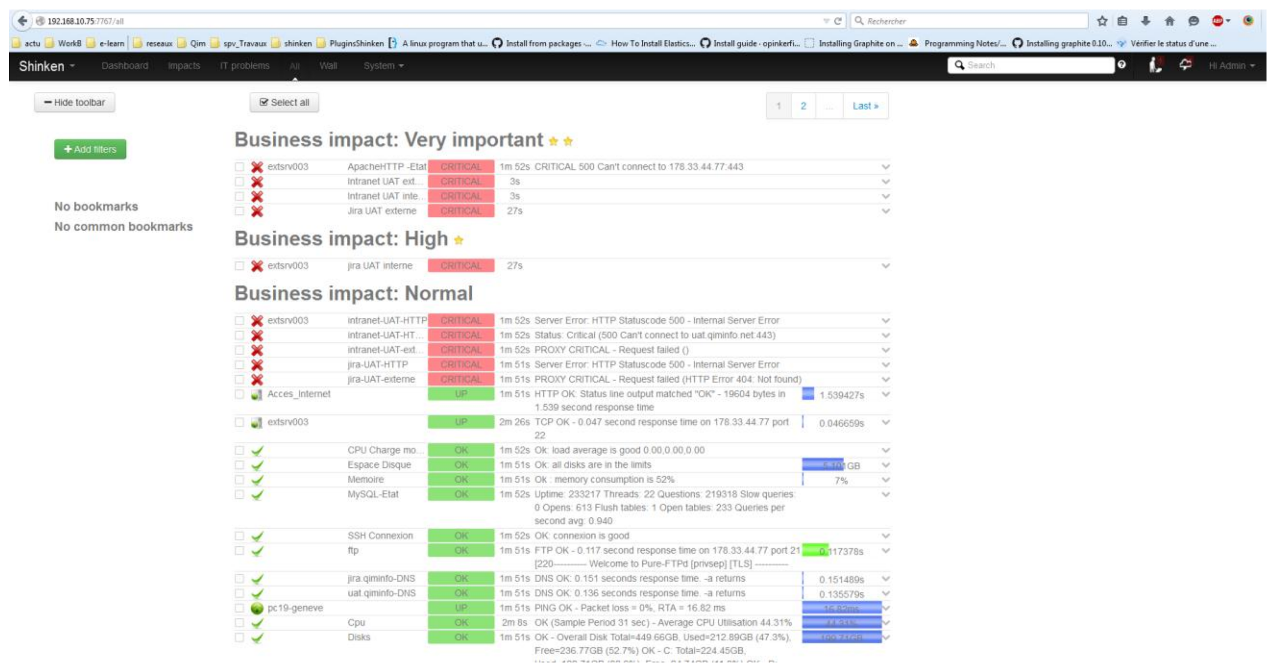
Date/Time: 10-06-2015 12:31:28
Additional Info : OK (Sample Period 83 sec) - Average CPU Utilisation 55.30%

** Figure 111 Mail reçu pour la résolution du problème*

c. Tests sur serveur extsrv003

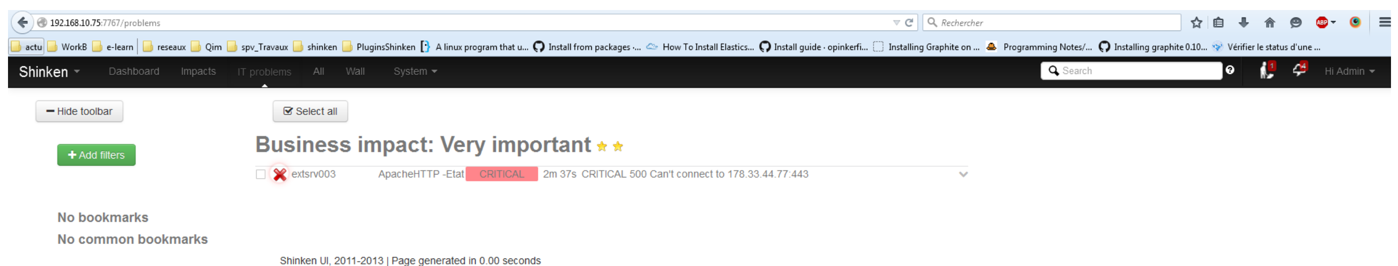
Tests serveur Apache

Dans la vue All on voit tous les problèmes lié ou pas au serveur Apache.



* Figure 112 Vue all pour le problème serveur Apache

Par contre dans la vue IT problems Shinken détecte un seul problème car les autres problèmes sont liés au



prblème du serveur Apache.

* Figure 113 La vue IT problems pour problème serveur Apache

La vue Walle affiche le business touché et la sources des problèmes.

* Figure 114 Vu Wall pour problème serveur Apache

Sous la vue Impacts on peut les dépendances du business et les problème liés.

* Figure 115 Vue Impacts pour problème serveur Apache

La vue currenlty.

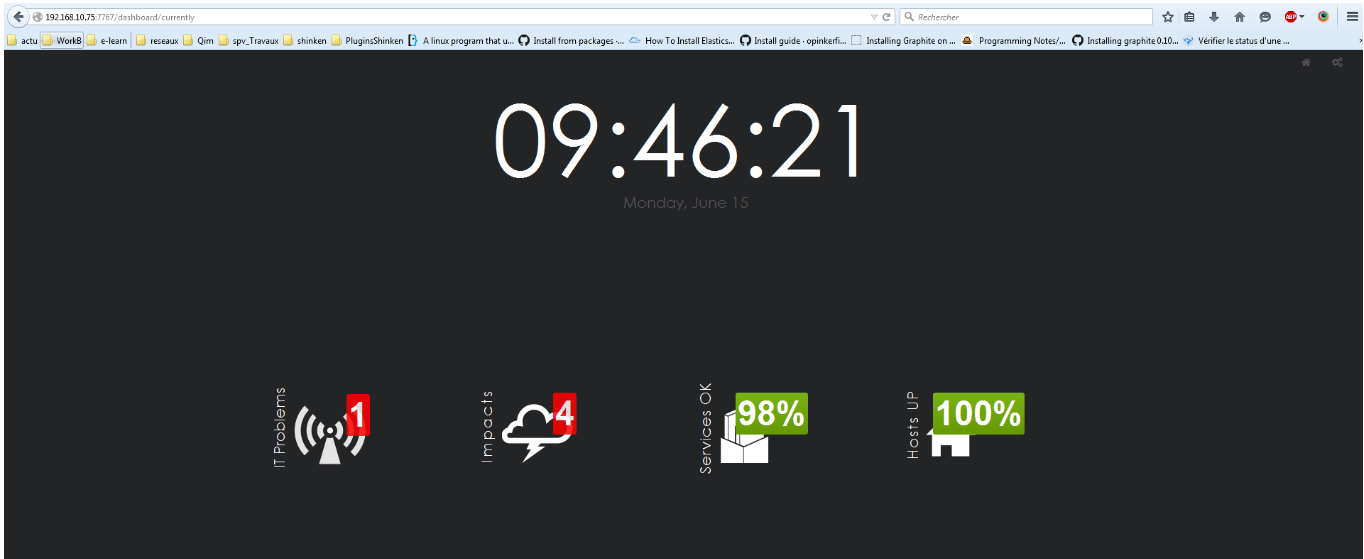


Figure 116 Vu Currently pour problème serveur Apache

Les mails recus. On recoit des notifications pour les règles métier aussi.

shinken		** RECOVERY alert - extsrv003/Intranet UAT externe is OK ** - Shinken Notification Notification Type: RECOVERY Service: Intranet UAT externe Host: extsrv003	09:48
shinken (2)	Shinken	** RECOVERY alert - extsrv003/jira UAT interne is OK ** - Shinken Notification Notification Type: RECOVERY Service: jira UAT interne Host: extsrv003 Address:	09:48
shinken (2)	Shinken	** RECOVERY alert - extsrv003/Jira UAT externe is OK ** - Shinken Notification Notification Type: RECOVERY Service: Jira UAT externe Host: extsrv003 Address:	09:48
shinken		** RECOVERY alert - extsrv003/Intranet UAT interne is OK ** - Shinken Notification Notification Type: RECOVERY Service: Intranet UAT interne Host: extsrv003	09:48
shinken		** RECOVERY alert - extsrv003/ApacheHTTP -Etat is OK ** - Shinken Notification Notification Type: RECOVERY Service: ApacheHTTP -Etat Host: extsrv003 Address:	09:47
shinken (3)		** PROBLEM alert - extsrv003/Intranet UAT externe is CRITICAL ** - Shinken Notification Notification Type: PROBLEM Service: Intranet UAT externe Host: extsrv003 Address	09:45
shinken (2)		** PROBLEM alert - extsrv003/Intranet UAT interne is CRITICAL ** - Shinken Notification Notification Type: PROBLEM Service: Intranet UAT interne Host: extsrv003 Address	09:44
shinken (3)	Shinken	** PROBLEM alert - extsrv003/jira UAT interne is CRITICAL ** - Shinken Notification Notification Type: PROBLEM Service: jira UAT interne Host: extsrv003 Address:	09:44
shinken (4)	Shinken	** PROBLEM alert - extsrv003/Jira UAT externe is CRITICAL ** - Shinken Notification Notification Type: PROBLEM Service: Jira UAT externe Host: extsrv003 Address:	09:44
shinken (2)		** PROBLEM alert - extsrv003/ApacheHTTP -Etat is CRITICAL ** - Shinken Notification Notification Type: PROBLEM Service: ApacheHTTP -Etat Host: extsrv003 Address:	09:44

Figure 117 Mails reçus pour problème serveur Apache

Détails du mail recu pour problème serveur Apache.

** PROBLEM alert - extsrv003/ApacheHTTP -Etat is CRITICAL ** Boîte de réception x

shinken@qgemnt001.localdomain Shinken Notification Notification Type: PROBLEM Service: ApacheHTTP -Etat Hos... 09:37 (Il y a 13 minutes)

shinken@qgemnt001.localdomain 09:43 (Il y a 6 minutes)
À moi

anglais > français Traduire le message Désactiver pour : anglais ×

Shinken Notification

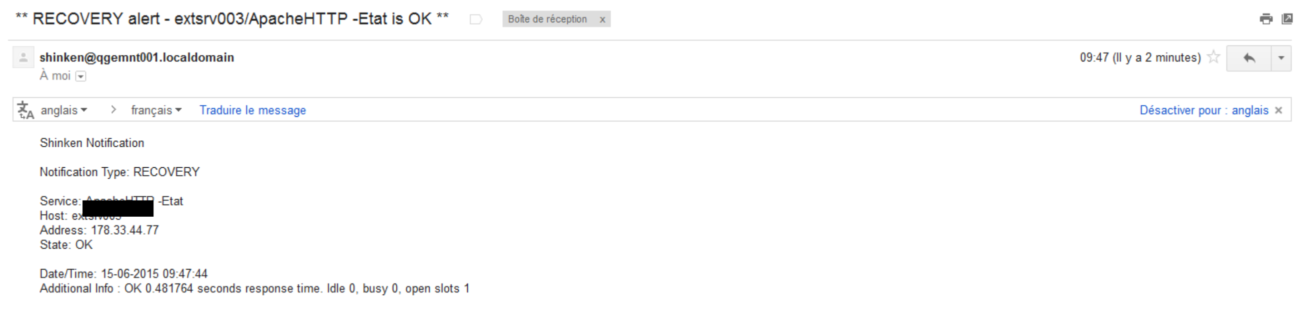
Notification Type: PROBLEM

Service: [redacted] -Etat
Host: ex[redacted]
Address: 178.33.44.77
State: CRITICAL

Date/Time: 15-06-2015 09:43:57

Figure 118 Détail du mail reçu pour problème serveur Apache

Détails du mail pour la résolution du problème du serveur Apache.


** RECOVERY alert - extsrv003/ApacheHTTP -Etat is OK **

Boîte de réception x

shinken@qgemnt001.localdomain

À moi

09:47 (Il y a 2 minutes)

anglais > français Traduire le message Désactiver pour : anglais

Shinken Notification

Notification Type: RECOVERY

Service: [redacted] -Etat
Host: [redacted]
Address: 178.33.44.77
State: OK

Date/Time: 15-06-2015 09:47:44
Additional Info : OK 0.481764 seconds response time. Idle 0, busy 0, open slots 1

Figure 119 Détail du mail de la résolution du problème serveur Apache

Annexe I. Captures des sondes via Wireshark

a. Sonde en SSH

	Time	Source	Destination	Protocc	Lengtl	Info
303	8.950450000	192.168.10.75	178.[redacted]	TCP	74	36979 > ssh [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=584358164 TSecr=0 WS=128
304	8.979974000	178.[redacted]	192.168.10.75	TCP	74	ssh > 36979 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1380 SACK_PERM=1 TSval=3879998670 TSecr=584358164 WS=128
305	8.980034000	192.168.10.75	178.[redacted]	TCP	66	36979 > ssh [ACK] Seq=1 Ack=1 Win=14720 Len=0 TSval=584358193 TSecr=3879998670
306	8.980814000	192.168.10.75	178.[redacted]	SSHv2	91	Encrypted request packet len=25
308	9.009357000	178.[redacted]	192.168.10.75	TCP	66	ssh > 36979 [ACK] Seq=1 Ack=26 Win=14592 Len=0 TSval=3879998700 TSecr=584358194
310	9.022053000	178.33.44.77	192.168.10.75	SSHv2	87	Encrypted response packet len=21
311	9.022085000	192.168.10.75	178.[redacted]	TCP	66	36979 > ssh [ACK] Seq=26 Ack=22 Win=14720 Len=0 TSval=584358235 TSecr=3879998712
315	9.023003000	192.168.10.75	178[redacted]	SSHv2	530	Client: Key Exchange Init
320	9.051194000	178.[redacted]	192.168.10.75	SSHv2	850	Server: Key Exchange Init
331	9.092178000	192.168.10.75	178.[redacted]	TCP	66	36979 > ssh [ACK] Seq=490 Ack=806 Win=16256 Len=0 TSval=584358306 TSecr=3879998741
332	9.092277000	178.[redacted]	192.168.10.75	TCP	66	ssh > 36979 [ACK] Seq=806 Ack=490 Win=15616 Len=0 TSval=3879998783 TSecr=584358236
333	9.092305000	192.168.10.75	178.[redacted]	SSHv2	338	Client: Diffie-Hellman Key Exchange Init
341	9.121758000	178.[redacted]	192.168.10.75	TCP	66	ssh > 36979 [ACK] Seq=806 Ack=762 Win=16640 Len=0 TSval=3879998812 TSecr=584358306
344	9.128216000	178.[redacted]	192.168.10.75	SSHv2	914	Server: New Keys
345	9.128250000	192.168.10.75	178.[redacted]	TCP	66	36979 > ssh [ACK] Seq=762 Ack=1654 Win=17920 Len=0 TSval=584358342 TSecr=3879998818
364	9.220080000	192.168.10.75	178.[redacted]	SSHv2	82	Client: New Keys
370	9.288049000	178.[redacted]	192.168.10.75	TCP	66	ssh > 36979 [ACK] Seq=1654 Ack=778 Win=16640 Len=0 TSval=3879998979 TSecr=584358433
371	9.288104000	192.168.10.75	178[redacted]	SSHv2	118	Encrypted request packet len=52
372	9.316754000	178.[redacted]	192.168.10.75	TCP	66	ssh > 36979 [ACK] Seq=1654 Ack=830 Win=16640 Len=0 TSval=3879999007 TSecr=584358501
373	9.316988000	178.[redacted]	192.168.10.75	SSHv2	118	Encrypted response packet len=52
374	9.317004000	192.168.10.75	178[redacted]	TCP	66	36979 > ssh [ACK] Seq=830 Ack=1706 Win=17920 Len=0 TSval=584358530 TSecr=3879999007
385	9.543920000	192.168.10.75	178.[redacted]	SSHv2	710	Encrypted request packet len=644
387	9.597587000	178.[redacted]	192.168.10.75	SSHv2	102	Encrypted response packet len=36
388	9.597633000	192.168.10.75	178.[redacted]	TCP	66	36979 > ssh [ACK] Seq=1474 Ack=1742 Win=17920 Len=0 TSval=584358811 TSecr=3879999288
389	9.626802000	192.168.10.75	178.[redacted]	SSHv2	134	Encrypted request packet len=68
390	9.661081000	178.[redacted]	192.168.10.75	SSHv2	118	Encrypted response packet len=52
391	9.670642000	192.168.10.75	178.[redacted]	SSHv2	134	Encrypted request packet len=68
393	9.707473000	178.[redacted]	192.168.10.75	SSHv2	154	Encrypted response packet len=88
394	9.728689000	178.[redacted]	192.168.10.75	SSHv2	186	Encrypted response packet len=120
395	9.728723000	178.[redacted]	192.168.10.75	SSHv2	138	Encrypted response packet len=72
398	9.747187000	192.168.10.75	178.[redacted]	TCP	66	36979 > ssh [ACK] Seq=1610 Ack=2074 Win=17920 Len=0 TSval=584358961 TSecr=3879999397
401	9.753603000	192.168.10.75	178.[redacted]	SSHv2	102	Encrypted request packet len=36
433	9.822420000	178.[redacted]	192.168.10.75	TCP	66	ssh > 36979 [ACK] Seq=2074 Ack=1646 Win=17920 Len=0 TSval=3879999513 TSecr=584358967
434	9.822513000	192.168.10.75	178.[redacted]	SSHv2	102	Encrypted request packet len=36
436	9.851296000	178.[redacted]	192.168.10.75	TCP	66	ssh > 36979 [ACK] Seq=2074 Ack=1682 Win=17920 Len=0 TSval=3879999542 TSecr=584359036
437	9.854395000	192.168.10.75	178.[redacted]	TCP	66	36979 > ssh [FIN, ACK] Seq=1682 Ack=2074 Win=17920 Len=0 TSval=584359068 TSecr=3879999542
441	9.883373000	178.[redacted]	192.168.10.75	TCP	66	ssh > 36979 [FIN, ACK] Seq=2074 Ack=1683 Win=17920 Len=0 TSval=3879999574 TSecr=584359068
442	9.883412000	192.168.10.75	178[redacted]	TCP	66	36979 > ssh [ACK] Seq=1683 Ack=2075 Win=17920 Len=0 TSval=584359097 TSecr=3879999574

Figure 120 Capture d'une sonde en SSH via Wireshark

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A→B	Bytes A→B	Packets A←B	Bytes A←B	Rel Start	Duration	bps A→B	bps A←B
192.168.10.75	36979	178.[redacted]	ssh	38	6 278	20	3 009	18	3 269	8.950450000	0.9330	25801.69	28031.15

** Figure 121 Taille d'une sonde SSH via Wireshark*

b. Sonde FTP

No.	Time	Source	Destination	Protocol	Length	Info
1080	43.518576000	192.168.10.75	178.[redacted]	TCP	74	52514 > ftp [SYN] Seq=0 Win=14600 Len=0 MSS=1460 SACK_PERM=1 TSval=1103958338 TSecr=0 WS=128
1084	43.621023000	178.[redacted]	192.168.10.75	TCP	74	ftp > 52514 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1380 SACK_PERM=1 TSval=1824579369 TSecr=1103958338 WS=128
1085	43.621069000	192.168.10.75	178.[redacted]	TCP	66	52514 > ftp [ACK] Seq=1 Ack=1 Win=14720 Len=0 TSval=1103958441 TSecr=1824579369
1097	43.692885000	178.[redacted]	192.168.10.75	FTP	385	Response: 220---------- Welcome to Pure-FTPd [privsep] [TLS] ----------
1098	43.692992000	192.168.10.75	178.[redacted]	TCP	66	52514 > ftp [ACK] Seq=1 Ack=320 Win=15744 Len=0 TSval=1103958513 TSecr=1824579470
1099	43.693163000	192.168.10.75	178.[redacted]	FTP	72	Request: QUIT
1100	43.693191000	192.168.10.75	178.[redacted]	TCP	66	52514 > ftp [FIN, ACK] Seq=7 Ack=320 Win=15744 Len=0 TSval=1103958513 TSecr=1824579470
1123	43.807814000	178.[redacted]	192.168.10.75	TCP	66	ftp > 52514 [ACK] Seq=320 Ack=7 Win=14592 Len=0 TSval=1824579540 TSecr=1103958513
1124	43.807854000	178.[redacted]	192.168.10.75	FTP	133	Response: 221-Goodbye. You uploaded 0 and downloaded 0 kbytes.
1125	43.807889000	192.168.10.75	178.[redacted]	TCP	54	52514 > ftp [RST] Seq=7 Win=0 Len=0
1126	43.807941000	178.[redacted]	192.168.10.75	TCP	66	ftp > 52514 [FIN, ACK] Seq=387 Ack=7 Win=14592 Len=0 TSval=1824579541 TSecr=1103958513
1127	43.807954000	192.168.10.75	178.[redacted]	TCP	54	52514 > ftp [RST] Seq=7 Win=0 Len=0

** Figure 122 Capture d'une sonde FTP via Wireshark*

Address A	Address B	Packets	Bytes	Packets A→B	Bytes A→B	Packets A←B	Bytes A←B	Rel Start	Duration	bps A→B	bps A←B
178.[redacted]	192.168.10.75	12	1 176	5	724	7	452	43.518576000	0.2894	20015.34	

** Figure 123 Capture la taille d'une sonde SNMP via Wireshark*

c. Sonde SNMP

La conversation entre le superviseur et routeur pour une sonde avec le SNMP.

No.	Time	Source	Destination	Protocol	Length	Info
1075	7.306990000	192.168.10.75	192.168.10.253	SNMP	93	get-next-request 1.3.6.1.2.1.2.2.1.2
1076	7.308528000	192.168.10.253	192.168.10.75	SNMP	150	get-response 1.3.6.1.2.1.2.2.1.2.2
1077	7.309948000	192.168.10.75	192.168.10.253	SNMP	94	get-next-request 1.3.6.1.2.1.2.2.1.2.2
1078	7.311509000	192.168.10.253	192.168.10.75	SNMP	145	get-response 1.3.6.1.2.1.2.2.1.2.3
1079	7.312625000	192.168.10.75	192.168.10.253	SNMP	94	get-next-request 1.3.6.1.2.1.2.2.1.2.3
1080	7.314506000	192.168.10.253	192.168.10.75	SNMP	145	get-response 1.3.6.1.2.1.2.2.1.2.4
1081	7.315673000	192.168.10.75	192.168.10.253	SNMP	94	get-next-request 1.3.6.1.2.1.2.2.1.2.4
1082	7.317503000	192.168.10.253	192.168.10.75	SNMP	145	get-response 1.3.6.1.2.1.2.2.1.2.5
1083	7.318629000	192.168.10.75	192.168.10.253	SNMP	94	get-next-request 1.3.6.1.2.1.2.2.1.2.5
1084	7.319617000	192.168.10.253	192.168.10.75	SNMP	145	get-response 1.3.6.1.2.1.2.2.1.2.6
1085	7.320806000	192.168.10.75	192.168.10.253	SNMP	94	get-next-request 1.3.6.1.2.1.2.2.1.2.6
1086	7.321622000	192.168.10.253	192.168.10.75	SNMP	145	get-response 1.3.6.1.2.1.2.2.1.2.7
1087	7.322723000	192.168.10.75	192.168.10.253	SNMP	94	get-next-request 1.3.6.1.2.1.2.2.1.2.7
1088	7.324494000	192.168.10.253	192.168.10.75	SNMP	145	get-response 1.3.6.1.2.1.2.2.1.2.8
1089	7.325550000	192.168.10.75	192.168.10.253	SNMP	94	get-next-request 1.3.6.1.2.1.2.2.1.2.8
1090	7.326341000	192.168.10.253	192.168.10.75	SNMP	145	get-response 1.3.6.1.2.1.2.2.1.2.9
1091	7.327507000	192.168.10.75	192.168.10.253	SNMP	94	get-next-request 1.3.6.1.2.1.2.2.1.2.9
1092	7.328628000	192.168.10.253	192.168.10.75	SNMP	145	get-response 1.3.6.1.2.1.2.2.1.2.10
1093	7.329721000	192.168.10.75	192.168.10.253	SNMP	94	get-next-request 1.3.6.1.2.1.2.2.1.2.10
1094	7.330512000	192.168.10.253	192.168.10.75	SNMP	150	get-response 1.3.6.1.2.1.2.2.1.2.11
1095	7.331625000	192.168.10.75	192.168.10.253	SNMP	94	get-next-request 1.3.6.1.2.1.2.2.1.2.11
1096	7.332478000	192.168.10.253	192.168.10.75	SNMP	152	get-response 1.3.6.1.2.1.2.2.1.2.12
1097	7.333627000	192.168.10.75	192.168.10.253	SNMP	94	get-next-request 1.3.6.1.2.1.2.2.1.2.12
1098	7.334400000	192.168.10.253	192.168.10.75	SNMP	144	get-response 1.3.6.1.2.1.2.2.1.2.13
1099	7.335514000	192.168.10.75	192.168.10.253	SNMP	94	get-next-request 1.3.6.1.2.1.2.2.1.2.13
1100	7.336445000	192.168.10.253	192.168.10.75	SNMP	140	get-response 1.3.6.1.2.1.2.2.1.2.14
1101	7.337518000	192.168.10.75	192.168.10.253	SNMP	94	get-next-request 1.3.6.1.2.1.2.2.1.2.14
1102	7.338334000	192.168.10.253	192.168.10.75	SNMP	141	get-response 1.3.6.1.2.1.2.2.1.2.15
1103	7.339415000	192.168.10.75	192.168.10.253	SNMP	94	get-next-request 1.3.6.1.2.1.2.2.1.2.15
1104	7.340302000	192.168.10.253	192.168.10.75	SNMP	95	get-response 1.3.6.1.2.1.2.2.1.3.2
1105	7.341749000	192.168.10.75	192.168.10.253	SNMP	94	get-request 1.3.6.1.2.1.2.2.1.8.3
1106	7.342508000	192.168.10.253	192.168.10.75	SNMP	95	get-response 1.3.6.1.2.1.2.2.1.8.3

** Figure 124 Capture d'une sonde SNMP via Wireshark*

Address A	Address B	Packets	Bytes	Packets A→B	Bytes A→B	Packets A←B	Bytes A←B	Rel Start	Duration	bps A→B	bps A←B
Vmware_72:e2:b7	Cisco_e8:2e:81	32	3 730	16	1 503	16	2 227	7.306990000	0.0355	338532.58	

** Figure 125 Capture taille d'une sonde SNMP via Wireshark*

Annexe J. Figures

Annexe K. Tableaux